Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 Visual Studio Application Lifecycle Management Administering Team Foundation Server
 See Also Send Feedback

Welcome to the administration guide for Visual Studio Team Foundation Server 2012, also known as TFS. This is the best place to start looking for information about how to configure TFS, manage it, and use it to support your business. This information is designed to help novice administrators learn about TFS, as well as help experienced administrators get the most out of this latest version.

Administrators of TFS install, configure, and maintain the deployment, and they perform the following tasks, among others:

- Back up and restore data
- Add servers, features and other resources to the deployment
- Administer permissions and security
- Organize projects and the resources that support them by creating and maintaining team project collections
- Maintain the security of the deployment by creating server-level and collection-level groups for managing users and their permissions
- Manage groups and permissions on servers that are integrated with the deployment, such as SQL Server Reporting Services and SharePoint Products, or coordinate with the administrators of those servers to ensure the smooth operation of Team Foundation Server with those resources
- Manage the services and service accounts that are used in the deployment

As an administrator of TFS, you should be familiar with Windows security

restrictions, user account control, and the operation of and security for SQL Server databases. If your deployment is integrated with SharePoint Products, you should also know how to manage site collections and Web applications.

If you are the administrator for a new or not-yet-installed deployment, you should carefully consider which features of TFS you want to install and how you want to install them. By choosing a deployment topology that best meets the needs of your business, you can help maintain the long-term manageability and sustainability of the software projects that you support. You can review the example topologies of TFS to see what kind of topology might best suit your needs, or you can follow the tutorial that helps you <u>Get Up and Running With a Single-Server Installation [Tutorial]</u>. You might decide that <u>Team Foundation Service</u>, a hosted version of TFS, is the best answer for you and your business.

There are both online and offline versions of both this guide and the installation guide:

- To find the most recent version of this guide, see this page on the Microsoft Web site: <u>Administering Team Foundation Server</u>. To download the most recent version of this guide for offline use, see this page on the Microsoft Web site: <u>Downloadable Administration Guide</u>.
- To read the most recent information about how to install Team Foundation and to find a downloadable version of the install guide, see this page on the Microsoft Web site: Installing Team Foundation Server.
- The offline versions of these guides are compiled help module (.chm) files. If you experience problems opening a downloaded .chm file, see the last section of this topic.

## ■Resources that help you administer TFS

Content	Common Tasks
Tutorials for Team Foundation ServerUnderstanding Administrative Concepts and ToolsOrganizing Your Server with Team Project CollectionsConfiguring Resources to Support Team ProjectsChange access levels	Learn about Team Foundation Server: You can review the features and capabilities of TFS, explore the server architecture and example deployment topologies, choose the deployment structure that best suits your business needs, and follow tutorials designed to help you get started with a single-server deployment.
Installing Team Foundation	Install Team Foundation Server: You can choose the

installation or upgrade

Server <u>Get Up and</u> <u>Running With</u> <u>a Single-</u> <u>Server</u> <u>Installation</u> [Tutorial]	scenario for your deployment, install prerequisites, and install the components of Team Foundation to support your software development projects.
To upgrade your edition of Team Foundation Server, see TFS Upgrade Requirements.	Upgrade Team Foundation Server You can upgrade from one edition of TFS to another. You can also upgrade from an earlier version of TFS to Visual Studio Team Foundation Server 2012.
Upgrading from Visual SourceSafe	Upgrade from Visual SourceSafe: You and your team can enjoy many benefits by upgrading your code projects, files, version history, labels, and user information from Visual SourceSafe to Team Foundation Server (TFS) version control.
Managing the Server Configuration	

Configuring Your Server Using the Team Foundation Administration Console	
Accounts and Dependencies in Team Foundation Server Backing up and Restoring Your Deployment	Manage the servers and resources in the deployment: After you install and initially configure TFS, you can monitor and change the configuration to support your development projects
Restoring a Single-Server Deployment to New Hardware [Tutorial] Stop and Start Services	development projects or increase the security of your deployment. • You should also create a backup plan to protect your data. You can also expand or t move portions of the deployment to better suit your operational needs.
<u>Application</u> <u>Pools, and</u> <u>Websites</u>	
Support alerts and feedback requests with an SMTP server Securing Team	

#### Foundation Server

<u>Create a Team</u> <u>Project</u> <u>Collection</u> <u>Modify a</u> <u>Team Project</u> <u>Collection</u>	Create team project collections: You can organize a group of development projects and the resources that support them into a team project collection. You can create as many team project collections as you require to support development efforts.
<u>Configuring</u> <u>and</u> <u>Administering</u> <u>Lab</u> Management	Add virtual machines as development and testing resources: You can configure Visual Studio Lab Management to support developing and testing your team's code on virtual environments.
Configure features after a TFS upgrade <u>Configuring</u> <u>Resources to</u>	Add reporting or project portal resources to existing team projects: You can add or change the resources that are available to members of your team. These resources can

<u>Support Team</u> <u>Projects</u>	include process guidance, a team project portal, and a shared document library.
Administering Team Foundation Version Control	Change how source code is stored and accessed: You can help ensure high-quality code by configuring check-in policies, configuring check-out settings, and optimizing access to the source code. (We are not currently republishing this section. However, you can read the Visual Studio 2010 version of this guidance.)
<u>Configuring</u> and Managing <u>Your Build</u> <u>System</u>	Automatically and consistently build, test, and deploy your app in a distributed environment: You can configure the build system components your team needs to use Team Foundation Build As your team and your code base grow, you can scale out and manage your build system.

Manage users and groups: You can control what users have access to team projects, team project collections, and the servers that run the components of Team Foundation. You can also control the set of tasks that each type of user can perform.

Configure Team Foundation Server to Support Your Development Teams

Collaborate (dig deeper)

Change permission levels

<u>Add Users to</u> <u>Team Projects</u>

Set Administrator Permissions for Team Project Collections

<u>Set</u> <u>Administrator</u>  Manage groups and group membership: You can more easily organize and manage users in TFS by using the default groups and their associated permissions in a team project, in a team project collection, or at the server level. You can also create custom groups and grant them specific permissions that match the job requirements of users who work on your projects.

 Manage users outside of groups: You can add a user

Permissions	to TFS with
<u>for Team</u>	different
<u>Foundation</u>	permissions than
<u>Server</u>	those granted by
Team Foundation Server Permissions Team Foundation Server Default Groups, Permissions, and Roles	group membership. • Manage Permissions: You can view and change the permissions that are assigned to a user or group at the server level, the collection level, and the project level. By understanding the system
	permissions, you
	can better define
	custom groups and
	their permissions
	to meet the
	security needs of
	your deproyment.
<u>Technical</u> <u>Reference for</u> <u>Team</u> Foundation	Technical Reference: You can learn about the command-line tools with which you can manage your deployment.

<u>TFS</u>

#### Administration forum

Visual Studio ALM Rangers Solutions and Projects

<u>Team</u>
<u>Foundation</u>
Server Blog

<u>Technical</u> <u>Articles for</u>

Visual Studio

Application Lifecycle Additional Resources: Sometimes the best way to learn about TFS is to interact with your fellow administrators on forums, blogs, and by reading real-world deployment examples.

<u>Agile</u> <u>Development</u> <u>Showcase</u>

<u>Management</u>

<u>TFS on</u> <u>CodePlex</u>

# **Opening a downloaded .chm file**

On a computer that has security update 896358 installed, you cannot open .chm files that you download. For more information, see the following page on the Microsoft Web site: <u>You cannot open HTML Help files from Internet Explorer after you install security update 896358 or Windows Server 2003 Service Pack 1</u>.

To work around this problem, use either of these methods:

## Method 1

- 1. Open the .chm file.
- 2. In the Open File-Security Warning dialog box, clear the Always ask before opening this file check box.
- 3. Choose Open.

## Method 2

- 1. Open the context menu of the .chm file, and then choose Properties.
- 2. Choose Unblock.
- 3. Open the .chm file.

## See Also

#### Concepts

Planning and Tracking Projects Improving Quality with Visual Studio Diagnostic Tools What's New for Application Lifecycle Management in Visual Studio 2012

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Tutorials for Team Foundation Server
See Also Send Feedback

Installing and managing a deployment of Team Foundation Server over time can seem intimidating, particularly if you are new to administering Team Foundation Server. However, you can review the following tutorials, which will help you understand the installation and configuration process, guide you through the steps required to create team projects and add users to them, and describe how to back up and restore your deployment for this pre-release version of Team Foundation Server. Even if your chosen deployment configuration does not exactly match the one depicted in the examples in these tutorials, these topics can still help guide you through the process of configuring and managing your own deployment.

#### ⊠Note

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in these tutorials are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

## In this section

Title

Description

configure, and manage a

Learn how to install,

Get Up and Running With a Single-Server Installation [Tutorial]

deployment of Team Foundation Server in a single-server configuration, including SQL Server Reporting Services and SharePoint Foundation 2010. Discover how to configure scripts to back up your single-server deployment in case of a hardware failure or other disaster.

Restoring a <u>Deployment</u> to New Hardware [Tutorial]

made of the Team **Single-Server** Foundation Server databases and of the SharePoint Foundation 2010 farm to restore the deployment to a new server in the event of an original hardware failure.

Learn how to use the groups you configured and the backups you

<u>Add reports</u> <u>to a team</u> <u>project</u> Learn how to add a report server to your TFS (on-premises) deployment if you initially configured TFS as a basic installation without reporting. You can then add reporting to team project collections and team projects. Reports provide insight to help teams manage work and improve processes.

Configuring Lab Management for SCVMM Environments

configure Lab Management for Team Foundation Server, including how to create and manage System Center Virtual Machine Manager (SCVMM) environments ad how to install a test controller.

Learn how to install and

## **See Also**

#### Concepts

Configure features after a TFS upgrade Collaborate (dig deeper) Collaborate Choose the Team Foundation client to support your tasks Installing Team Foundation Server <u>Understanding Administrative Concepts and Tools</u> <u>Configuring Users, Groups, and Permissions</u> <u>Technical Reference for Team Foundation</u>

#### **Other Resources**

Process Guidance and Process Templates for Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Get Up and Running With a Single-Server Installation [Tutorial]
Send Feedback

Installing Team Foundation Server is fairly straightforward. However, managing that installation can become complex. As an administrator, you need to create projects, add users to the system, and set permissions for those users. You have to manage all the users in your system to help keep your deployment secure from those who shouldn't have access to it, but readily available and functional for those who should. You also want to make sure that your deployment is backed up just in case something happens to the server hardware, or some other disaster occurs, such as a virus that corrupts your data.

Preparing for your installation ahead of time can make it simpler to manage users over the lifetime of the deployment. Similarly, setting up backups for the system as soon as you have installed and configured the software can help you recover your deployment if it becomes necessary.

#### ⊠Note

This tutorial contains fictitious users, servers, and companies as part of the examples that help illustrate the procedures. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

# **Tutorial contents**

Title	Description
Prepare for Installation	Learn about users, groups, and permissions in Team Foundation Server deployments, and plan ahead to make managing these permissions simpler across a deployment that uses SQL Server Reporting Services, SharePoint Foundation 2010, and Team Foundation Server. Learn how to create and manage groups of users and create accounts to use as service accounts.
<u>Install</u> <u>Prerequisites</u> <u>and Team</u> <u>Foundation</u> <u>Server</u>	Learn how to install Team Foundation Server in one of the most common installation types. You'll learn how to install the prerequisites for a single-server installation, and to install Team Foundation Server by using the Standard Wizard.

Configure Team Foundation Server to Support Your Development Teams	Learn how to create projects in Team Foundation Server. You'll learn how to configure groups of users to those projects so that they have the permissions they need in Team Foundation Server, SQL Server Reporting Services, and SharePoint Foundation 2010. Also, learn how to restrict a group of users from viewing a project in Team Foundation Server.
<u>Create Back</u> Up Schedule	Learn how to manually configure backups for all of the components used

## <u>Up Schedule</u> and Plan

configure backups for all of the components used by a single-server deployment of Team Foundation Server.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Prepare for Installation
Send Feedback

This topic, the first part of the Single-Server Installation tutorial, teaches you how to create groups to manage users across Team Foundation Server, SharePoint Foundation 2010, and SQL Server Reporting Services. You'll also learn what kinds of accounts can be used as service accounts in the deployment, so that you can create these accounts, if necessary, before beginning the installation process.

The examples in this tutorial topic follow Jill Frank, an IT Administrator at the fictitious Fabrikam Fiber company, as she installs and deploys Team Foundation Server in a single-server configuration in order to support her software development teams.

In this topic

- <u>Users, groups, and permissions in Team Foundation Server deployments</u>
  - To create a group in Active Directory
  - <u>To create a local group on the server</u>
  - To create an account to use as a service account in Active Directory
  - To create a local account to use as the service account on the server

Required permissions

To perform the procedures in this tutorial, you must be a member of these groups:

• The Administrators security group on the server where you intend to install

Team Foundation Server.

• The Account Operators, Domain Admins, or Enterprise Admins group, or have the equivalent permissions in Active Directory Domain Services, if you intend to create Active Directory groups to manage the users in your deployment, or use an Active Directory account as a service account. Also, if you intend to use managed service accounts as part of the deployment, you might require further permissions and additional software to create that managed service account. For more information, see <u>Service Accounts</u> <u>Step-by-Step Guide</u>.

# **■Users, groups, and permissions in** Team Foundation Server deployments

Team Foundation Server, SharePoint Products, and SQL Server Reporting Services all maintain their own information about groups, users, and permissions. To make managing users and permissions across these programs simpler, you can create groups of users with similar access requirements in the deployment, give those groups appropriate access in the different software programs, and then just add or remove users from a group as needed. This is much easier than maintaining individual users or groups of users separately in three separate programs.

If your server is in an Active Directory domain, one option is to create specific Active Directory groups to manage your users, like a group of developers and testers for all projects in the team project collection, or a group of users who can create and administer projects in the collection. Similarly, you can create an Active Directory account for services that can't be configured to use the Network Service system account as the service account. To do so, create an Active Directory account for SharePoint Foundation 2010 and as the read-access data source account for reports in SQL Server Reporting Services.

#### ĭTip

In the example deployment, Jill chooses to use the Network Service system account as the service account for Team Foundation Server and SQL Server 2008. This is the default choice. If you want to use a specific account as the service account for security purposes or other reasons, you can. Jill also creates a specific Active Directory account to use as the service account for SharePoint Foundation 2010 and as the data source reader account for SQL Server Reporting Services. She names this account SVCSPTRS.

If your server is in an Active Directory domain but you don't have permissions to create Active Directory groups or accounts, or if you're installing your server in a workgroup instead of a domain, you can create and use local groups to manage

users across SQL Server 2008, SharePoint Foundation 2010, and Team Foundation Server. Similarly, you can create a local account to use as the service account. However, keep in mind that local groups and accounts are not as robust as domain groups and accounts. For example, in the event of a server failure, you would need to recreate the groups and accounts from scratch on the new server. If you use Active Directory groups and accounts, the groups and accounts will be preserved even if the server hosting Team Foundation Server fails.

In the example deployment, Jill analyzes the business requirements for the new deployment and reviews the security requirements with the project managers. She decides to create three groups to manage the majority of users in the deployment:

- A general group for developers and testers who will participate fully in all projects in the default team project collection. This group will contain the majority of users. She names this group TFS\_ProjectContributors, which resolves to the friendly name "Fabrikam Developers and Testers."
- A small group of project administrators who will have permissions to create and manage projects in the collection. She names this group TFS\_ProjectAdmins, which resolves to the friendly name "Fabrikam Project Administrators."
- A special, restricted group of contractors who will only have access to one of the projects. She names this group TFS\_RestrictedAccess, which resolves to the friendly name "Fabrikam Limited Access."

Later on, as the deployment expands, she might decide to create other groups.

## To create a group in Active Directory

• Create a security group that is a local domain, global, or universal group in Active Directory, as best meets your business needs. For example, if the group needs to contain users from more than one domain, the universal group type will best suit your needs. For more information, see

#### Create a New Group (Active Directory Domain Services).

In the example deployment, Jill creates three Active Directory groups, as

specified above.

### To create a local group on the server

• Create a local group and give it a name that will quickly identify its purpose. By default, any group you create will have the equivalent permissions of the Users default group on that computer. For more information, see <u>Create a local group</u>.

# To create an account to use as a service account in Active Directory

• Create an account in Active Directory, set the password policy according to your business requirements, and make sure that Account is trusted for delegation is selected for the account. For more information, see <u>Create a New User Account (Active Directory Domain Services)</u> and <u>Understanding User Accounts (Active Directory Domain Services)</u>.

In this tutorial, Jill uses the system account Network Service as the service account for Team Foundation Server, but she creates a specific Active Directory account to use as the service account for SharePoint Foundation 2010 and as the data source reader account for SQL Server Reporting Services. She names this account SVCSPTRS.

# To create a local account to use as the service account on the server

• Create a local account to use as the service account and then modify its group membership and other properties according to the security requirements for your business. For more information, see <u>Create a local user account</u>.

Home | Prepare for Installation | Install Prerequisites and Team Foundation Server | Configure Team Foundation Server to Support Your Development Teams | Create Back Up Schedule and Plan

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Install Prerequisites and Team Foundation Server
Send Feedback

This second topic in the Single-Server Installation tutorial teaches you how to install and configure the prerequisites required by Team Foundation Server, as well as how to install Team Foundation Server in a single-server configuration that uses SharePoint Foundation 2010 and SQL Server Reporting Services.

The examples in this tutorial topic follow Jill Frank, an IT Administrator at the fictitious Fabrikam Fiber company, as she installs and deploys Team Foundation Server in a single-server configuration in order to support her software development teams.

Installing Team Foundation Server and its prerequisites is easier than you might think. While all software installation requires some time and patience, you can follow the steps below to streamline the process and install a new single-server deployment of Team Foundation Server, complete with reporting provided by SQL Server Reporting Services and team project portals hosted on SharePoint Foundation 2010.

#### ĭTip

There are many different ways you can install and configure Team Foundation Server. This tutorial shows you how to install a new server using all new installations of the prerequisites, but you can choose to use existing installations of the prerequisites, or to install Team Foundation Server in a different configuration. For more information, see Installing Team Foundation Server.

In the example deployment, Jill reviewed the system requirements and various configuration types for Team Foundation Server by reading the installation guide. She discussed the projected number of users and projects with the project

managers, and the features those managers were most interested in. After reviewing the choices and the requirements, Jill decided that the best configuration for her software development teams and the business requirements of the project managers was to install Team Foundation Server on a single server using the Standard Wizard. The Standard Wizard configuration is relatively simple, and uses the Network Service account as the service account for Team Foundation Server. The wizard also installs SharePoint Foundation 2010, which requires a separate service account. In the previous topic in this tutorial, <u>Prepare for Installation</u>, Jill created an Active Directory account specifically for this purpose.

Before Jill can install Team Foundation Server, she must first install and configure the prerequisites on her server, which is running the 64-bit edition of the Windows Server 2008 R2 operating system with Service Pack 1 installed, is equipped with a dual-core processor and 10 GB of RAM, and has 300 GB of free disk space. In order to meet the prerequisites for Team Foundation Server, she must install a supported version of SQL Server to support both Team Foundation Server and SharePoint Foundation 2010. She chooses to install SQL Server 2008 R2, Standard Edition.

In this topic

- Installing SQL Server 2008 R2 to support Team Foundation Server
- Installing Team Foundation Server on your server
- Installing Team Explorer on your server

#### **Required permissions**

To perform the procedures in this topic, you must be a member of the Administrators security group on the server where you want to install the software.

# To install SQL Server 2008 R2 to support Team Foundation Server

1. Launch the SQL Server Installation Center.

- 2. On the SQL Server Installation Center page, choose Installation, and then choose New installation or add features to an existing installation.
- 3. On the Setup Support Rules page, verify that all rules have passed, and then choose OK.
- 4. On the Product Key page, provide your product key or choose a free edition, and then choose Next.

#### ĭTip

If you choose an evaluation edition and do not update it before the evaluation expires, your deployment of Team Foundation Server will become unusable.

- 5. On the License Terms page, review the license agreement. If you accept the terms, select I accept the license terms. Optionally select the check box to send usage data to Microsoft, and then choose Next.
- 6. On the Setup Support Files page, choose Install.
- 7. On the Setup Support Rules page, review the setup information. Correct any failure conditions, and then choose Next.
- 8. On the Setup Role page, choose SQL Server Feature Installation, and then choose Next.
- 9. On the Feature Selection page, select the following check boxes, and then choose Next:
  - Database Engine Services
  - Full-Text Search
  - Analysis Services
  - Reporting Services
  - Client Tools Connectivity

- Management Tools Basic
- Management Tools Complete



#### ĭTip

While the management tools are not technically required for Team Foundation Server, they're required for creating the backup scripts you'll need to back up your deployment in this release, and they make the management of your deployment much simpler.

- 10. On the Installation Rules page, address any failures or warnings, and then choose Next.
- 11. On the Instance Configuration page, choose Default instance, and then choose Next.
- 12. On the Disk Space Requirements page, review the information to make sure you have sufficient disk space, and then choose Next.
- On the Server Configuration page, choose Use the same account for all SQL Server services. In the Use the same account for all SQL Server 2008 R2 services window, choose or specify NT AUTHORITY\NETWORK SERVICE, and then choose OK.

In the Startup Type column, specify Automatic for all services that you can edit, and then choose Next.

Service Accounts	Collation				
dicrosoft recommend	ls that you use a se	eperate account for each SQL S	erver service.		
Service		Account Name	Password	Startup Typ	e
SQL Server Agent		NT AUTHORITY\NETWO		Automatic	•
SQL Server Database	Engine	NT AUTHORITY\NETWO		Automatic	
SQL Server Analysis	Services	NT AUTHORITY\NETWO		Automatic	•
SQL Server Reporting	g Services	NT AUTHORITY\NETWO		Automatic	-
SQL Full-text Filter D	aemon Launcher	NT AUTHORITY\LOCAL S		Manual	
SQL Server Browser		NT AUTHORITY\LOCAL S		Automatic	-

- 14. On the Database Engine Services page, on the Account Provisioning tab, choose Windows authentication mode and then choose Add Current User to add your account as an administrator for this instance of SQL Server. Optionally, add any other user accounts for users you want to act as database administrators, and then choose Next.
- 15. On the Analysis Services Configuration page, on the Account Provisioning tab, choose Add Current User to add your account as an administrator for the analysis services database. Optionally, add any other user accounts for users you want to act as administrators, and then choose Next.
- 16. On the Reporting Services Configuration page, choose Install the native mode default configuration, and then choose Next.
- 17. On the Error Reporting page, choose whether to send information about errors to Microsoft, and then choose Next.
- 18. On the Installation Rules page, review any failures or warnings, and then choose Next.
- 19. On the Ready to Install page, review the list of components to be installed, and if they match the list of features shown in the illustration below, then choose Install. If you need to make any changes, choose Back.



On the Installation Progress page, optionally monitor the installation progress of each component. When all components have finished installing, the Complete page appears.

- 20. On the Complete page, choose Close.
- 21. After you finish the installation, visit Microsoft Update to apply any needed service packs or updates that have been released for your product. You must install SP1 in order to meet the basic requirements for Team Foundation Server.

# Installing Team Foundation Server on your server

Once you've finished installing the prerequisites, you're ready to install Team Foundation Server itself. In the example deployment, Jill has decided to install Team Foundation Server using the Standard Wizard.

### To install Team Foundation Server using the Standard Wizard

1. Launch the installation media. In Team Foundation Server Configuration Center, choose Standard Single Server, and then choose Start Wizard.

The Team Foundation Server Standard Configuration wizard appears.

Configuration Center				
Configuration Team Foundation Application Server Basic Standard Single Server Advanced Application-Tier Only Upgrade Configure Team Foundation Server Proxy Configure Team Foundation Build Service Configure Extensions for SharePoint Products	About this Wizard Use the Standard wizard to install Team Foundation Server on a This wizard is suitable for most single-server deployments. Supported on Windows Server operating systems only You might want to use this wizard if one or more of the fol • You want to deploy Team Foundation Server on a single ser • You want to deploy Team Foundation 2010 installed and • You want to use the default instance of SQL Server. You do not want to use thi wizard if on or more of the follow • A version of SharePoint Products is already installed on this • You want to utilize remote servers for SQL Server databases			

- 2. On the Welcome page, choose Next.
- 3. On the Service Account page, in User Account, specify the name of the service account that you want to use for SharePoint Foundation 2010 and as the data source report reader account in SQL Server Reporting services.

In Password, specify the password for this service account.

Optionally choose Test to test if the service account is a valid account. This step is not required, but is recommended. When you have finished providing all the information for the account, choose Next.

Standard Configuration Wizard				
Welcome       Provide a Service Account to U         Installation       Installation         Review       Review         Readiness Checks       Review		vice Account to Use for SharePoint Products and the default system account Network Service for the services used tem account is required for use as the service account for SharePoi der account that is used for accessing SQL Server Reporting Serv word below.		
Complete	Account Name: Password:	MyServiceAccount		

- 4. On the Installation page, review the information and any warnings, and then choose Install SharePoint Foundation 2010.
- 5. After Team Foundation Server completes the installation of SharePoint Foundation 2010, a success message appears. Choose Next.
- 6. On the Review page, review the information for the installation of Team Foundation Server. If any changes are needed, choose Previous, and make your changes. Once all the information is correct, choose Next.

Standard Configuration Wizard		
Welcome Service Account Installation	Service Account Authentication Method Configure IIS Site URL	NT Authority\NetworkService NTLM True http://FABRIKAMPRIME:8080/tfs
Review	Host name	FABRIKAMPRIME
Readiness Checks	Virtual Directory	tfs
Configure	Web Access URL	http://FABRIKAMPRIME:8080/tfs
Complete	Configuration Database	
	SQL Instance	FABRIKAMPRIME
	Use Existing Database	False
	Create Configuration Database	True
	Database Label Configuration Database	Tfs_Configuration

7. On the Readiness Checks page, the wizard will verify that all components are ready for installation. If any of the readiness checks fail or provide warnings, review the information and make any adjustments as necessary.

#### ĭTip

While a failed readiness check will prevent you from installing the software until the issue is addressed, a warning will not prevent you from proceeding with the installation.

When you are satisfied with the readiness checks, choose Configure.

- 8. On the Configuration Progress page, optionally monitor the progress of the installation. When the installation completes, choose Next.
- 9. Choose Close to close the Standard Configuration Wizard.
- 10. Choose Close to close the Configuration Center.

The Team Foundation Server Administration Console opens. Review the information for your newly configured server.

While Team Explorer is not necessary for most administration tasks, it does have

some administrative features, and it is the only way to create the team projects your software development teams will need. Installing it on the server is a convenient way to ensure that you can create projects whenever necessary, without having to go to another computer.

## **To install Team Explorer**

- 1. Browse the installation media for Team Foundation Server, and open the Team Explorer folder.
- 2. Find the vs\_TeamExplorer application, and run it.

The Visual Studio Team Explorer installation window opens.

3. Select I agree to the License terms and conditions, and then choose Install.

You are ready to start creating your first team projects!

Home |

<u>Prepare for Installation | Install Prerequisites and Team Foundation Server |</u> <u>Configure Team Foundation Server to Support Your Development Teams |</u> <u>Create Back Up Schedule and Plan</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Configure Team Foundation Server to Support Your Development Teams
Send Feedback

This third part of the Single-Server Installation tutorial teaches you how to create team projects using one of the process templates included with Team Foundation Server. You'll also learn how to add Active Directory users and groups to appropriate groups in Team Foundation Server, SharePoint Foundation 2010, and SQL Server Reporting Services, including how to modify permissions for those security groups, if needed.

The examples in this tutorial topic follow Jill Frank, an IT Administrator at the fictitious Fabrikam Fiber company, as she installs and deploys Team Foundation Server in a single-server configuration in order to support her software development teams.

Installing the software is only part of an administrator's job. Another equally critical part of managing a deployment is making sure that the users in the deployment have appropriate access to the resources in that deployment. You can use the groups you created earlier in this tutorial to help minimize the amount of overhead required to manage your users, and also help ensure that your users have the appropriate level of access to the features and projects in Team Foundation Server.

In this topic

- <u>Creating and configuring team projects</u>
- <u>Restricting access to projects in the deployment</u>
- <u>Configuring access to features in Team Web Access</u>

**Required Permissions** 

To perform the procedures in this tutorial, you must be a member of the following groups:

- The Administrators security group on the server on which you intend to install Team Foundation Server
- The Team Foundation Administrators group in Team Foundation Server
- The SQL Server System Administrator security group in SQL Server
- The Farm Administrators group in SharePoint Foundation 2010
- If you are using and configuring Active Directory group membership, you must be a member of Account Operators, Domain Admins, or Enterprise Admins, or have the equivalent permissions in Active Directory Domain Services.
# Creating and configuring team projects

A team project is the fundamental unit in Team Foundation Server. Each team project that is hosted within a team project collection is a separate unit with its own separate groups, permissions, and work items. You can use the project to store your source code, file bugs, feature requests, and other work items, and coordinate work across teams, schedules, and functional areas. However, the work item types available to each project will depend on the process template you choose when you create a team project. For this release, Team Foundation Server includes three process templates: MSF for Agile Software Development, MSF for CMMI Process Improvement, and Microsoft Visual Studio Scrum.

In this example deployment, Jill has discussed the process template requirements with the project managers, and learned that the three project managers all want to use the Scrum process in their projects. She creates the project using that process template, and additionally chooses to create a source control repository for all three projects. She knows that because she has installed and configured SharePoint Foundation 2010 and SQL Server Reporting Services as part of her deployment, the projects will have team project portals created for them in SharePoint Foundation 2010, and reports will be automatically available for the projects. However, she also knows that she will have to configure user permissions in all three software programs.

## To create a team project

1. Open Team Explorer and connect to the default collection.

## ∎Tip

If you're running Team Explorer on Windows Server 2008 or Windows 7, you'll need to start Team Explorer by choosing the option to Run as administrator.

In the example deployment, on the Team menu, Jill chooses Team, and then chooses Connect to Team Foundation Server. Because Jill has never connected to the server before, the list of servers is blank. She chooses Servers, and in the Add/Remove Team Foundation Server dialog, she chooses Add. In Add Team Foundation Server, she types the name of the server, which is FabrikamPrime. She leaves the rest of the connection details set to the default settings, because that is what her deployment uses, and chooses OK. She makes sure that the server name appears in Add/Remove Team Foundation Server, and chooses Close to close that dialog. In Servers, she makes sure that FabrikamPrime appears in the Select a Team Foundation Server drop-down box, and that DefaultCollection appears in Team Project Collections, and then chooses Connect. The next time she opens Team Explorer, it will automatically connect to this server and collection.

2. In the Team Explorer sidebar, choose Home, choose Projects and My Teams, and then choose New Team Project.

The Create Team Project wizard opens.

3. On the Specify the Team Project Settings page, in What is the name of the team project text box, specify a name for the project that will easily identify it to the users who will work in that project. Optionally, provide a description of the project, and then choose Next.

## ĭTip

There are some restrictions on project names, such as character limits and restricted characters. For more information, see Restrictions for Project Names.

4. On the Select a Process Template page, choose a process template to use from the drop-down list, and then choose Next.

In this example deployment, Jill chooses Microsoft Visual Studio Scrum.

5. On the Team Site Settings page, make sure that Create a new SharePoint Site is specified, and then choose Next.

## ĭTip

If you didn't choose to install SharePoint Products as part of your deployment, you won't have this option.

- 6. On the Specify Source Control Settings page, make sure that the Create an empty source control folder is specified, and then choose Next.
- 7. On the Confirm Team Project Settings page, review your choices, and then choose Finish.

It might take a few minutes for the wizard to complete creating the project.

Jill repeats this process three times, creating the following projects:

- Phone Saver
- Fuel Tracker 2
- Fabrikam Fiber Internal Tools

After she creates these projects, she then adds the groups she created to Team Foundation Server, SQL Server Reporting Services, and SharePoint Foundation 2010. In the procedures below, Jill adds the group "TFS\_ProjectContributors"-which resolves to the friendly name Fabrikam Developers and Testers-at the appropriate permissions level for users who will contribute fully to the project. For more information about the various contribution levels, see

## Add Users to Team Projects.

## To add a group to the Contributors group in a team project

1. In Team Explorer, open the team project to which you want to add users, and then choose Settings.

The Settings menu appears.

2. In the Settings menu, in Team Project, choose Group Membership.

Team Web Access opens in the administration context for the team project, on the security tab.

3. On the groups tab, choose Contributors, and on the members tab, choose Add, and then choose Add user.

The ADD A WINDOWS USER OR GROUP window opens.

4. In Identities, specify the name of the group you want to add. In the example deployment, Jill created a group, called Fabrikam Developers and Testers, for the developers and testers who would work on all team projects in the collection. She types that name into the box and presses Enter.

To add a wii Domain\Use	ndows user or group that is not currently known to Team Fo rname. If the member is known, just type their display name	undatio a.	n Server, type their
Indentities	Fabrikam Developers and Testers		
		V	browse   check name

The first time you add a user or group to Team Foundation Server, you cannot browse to it or check the name. After the identity has been added anywhere in Team Foundation Server, you will be able to use those functions to quickly find the name.

5. When you have finished adding the groups or users you want to add, choose Save Changes.

## To add a group as Contributors to the team project site in SharePoint Foundation 2010

1. In Team Explorer, on the Team menu, choose Show Project Portal.

The portal for the team project opens in a separate window.

2. Choose Site Actions, and then choose Site Permissions.

The browser window changes to Permissions Tools.

3. Choose Grant Permissions.

The Grant Permissions window opens.

4. In Users/Groups, specify the name of the group that you want to add. In Grant Permissions, choose Grant users permission directly, select the Contribute check box, and then choose OK.

Select Users	Users/Groups:
You can enter user names, group names, or e-mail addresses. Seperate them with semicolons.	FABRIKAMPRIME\tfs project contributors ;
Grant Permissions Select the permissions you want these users to have. You can add users to a SharePoint group that has already been granted the appropiate permission levels, or you can grant the users specific permission levels. Adding users to a SharePoint group is recommended, as this makes managaging permissions easier across multiple sites.	<ul> <li>Grant Permissions</li> <li>Add users to a SharePoint group (recommended) <ul> <li>(none)</li> <li>Grant users permissions directly</li> <li>Full Control - Has full control.</li> <li>Design - Can view, add, update, delete, approve, and customize.</li> <li>Contribute - Can view, add, update, and delete list items and documents.</li> <li>Read - Can view pages and list items and download documents.</li> </ul> </li> </ul>

Unlike Team Foundation Server and SharePoint Foundation 2010, SQL Server Reporting Services does not distinguish between projects. Therefore, if you add a group to Reporting Services, that group will have the same permissions for reports across all the projects in the collection, regardless of their permissions in individual projects. Keep this in mind when choosing what groups to add.

## To add a group to SQL Server Reporting Services

1. Open Internet Explorer or another browser compatible with Reporting Services administration.

2. Type the following into the address bar, where ReportServer represents the name of the server that is running SQL Server Reporting Services:

#### http://ReportServer/Reports/Pages/Folder.aspx

For example, Jill types: http://FabrikamPrime/Reports/Pages/Folder.aspx into the address bar.

- 3. On the Home page, choose Folder Settings.
- 4. On the Security page, choose New Role Assignment.
- 5. On the New Role Assignment page, in Group or use name, specify the name of the group you want to add. Select the Browser check box, and then choose OK.

- Æ	Report Manager	- Windows Int	ernet Explorer	
				Home   My
-	SQL Se Hon	erver Report <b>1e</b>	ing Services	
		X Delete	👷 New Role Assignment	
	Security		Group or User 👃	Role(s)
		Edit	BUILTIN\Administrators	Content Manager
		🗖 Edit	FABRIKAMPRIME\TFS Project Contributors	Browser
		Edit	FABRIKAMPRIME\TFS ProjectAdmins	Team Foundation Content Manager

# ■Restricting access to projects in the deployment

Sometimes you don't want all users in your deployment to have visibility into all projects in that deployment. By default, users who have permissions to access one project within a collection can view other projects within that collection, even if they don't have permissions to modify work items or perform other actions in that project. If you want to restrict a particular group to just one project in the collection, you must take extra steps.

In Team Foundation Server, permissions explicitly set to Deny generally take precedence over permissions set to Allow, whether through inherited allow or through explicitly setting the permission to allow. There are exceptions to this, but they generally don't apply to user groups (and you can read more about these exceptions in

Team Foundation Server Permissions). So if you want to restrict a particular group from viewing a particular project, you must first create a specific Team Foundation Server group in that project, then add that restricted group to that project-level group, and then explicitly set the View project-level information permission to Deny for that Team Foundation Server group. In other words, you specifically create a group for the users you don't want to view a project, *add* that group to the project you don't want them to view, and then set permissions on that group to *restrict* the users in that group from viewing that project. It's a little counterintuitive, but it works!

## To restrict a group from viewing a project

- 1. Open Team Web Access, change views to the administration context for the project, and choose the security tab.
- 2. On the groups tab, choose Create TFS Group.

The CREATE NEW TEAM FOUNDATION SERVER GROUP window opens.

3. In Group Name, specify a name for this group. In the example deployment, Jill names this group "Restricted". Optionally, type a description for this group, and then choose OK.

The group you just created appears in the list of TFS Groups. Make sure that it is highlighted in the list, and then choose the members tab.

4. Choose Add user.

The ADD A WINDOWS USER OR GROUP window opens.

- 5. In Identities, specify the name of the group you want to add. In the example deployment, Jill created a group for the contract employees who are only working on one project in the collection. She types that name into the box and presses Enter.
- 6. When you have finished adding the groups or users you want to add, choose Save Changes.
- 7. Choose the permissions tab. In the permissions list, toggle the value of View project-level permission to deny, and then choose Save Changes.

## ■Configuring access to features in Team Web Access

Not all features of Team Web Access are available to your users by default. Access to Team Web Access features is controlled through the administration mode of Team Web Access. Default access level is set for the entire deployment, not for individual team project collections or groups. To configure access levels for specific users of collections, team projects, or teams, you must manually add those users or groups to one of the three access groups in Team Web Access. If you want your users to have access to all the features available, or if you want to restrict some users from accessing the default features available, you'll need to configure that access. You'll also want to make sure that you have the appropriate licensing in place to use those features. For more information, see the

Control Panel				
control panel	web access	extensions		
Export Audit Log		Web A	access Permissions	
Limited Standard (Default)		Name	Standard	
Full		Set as defa	ult web access   Add   Search	
		No ident	ities found in current scope.	

#### Team Foundation Server Licensing Whitepaper.

## To view or change the default feature access group of Team Foundation Server

1. Open Team Web Access in administration mode, and choose Control Panel to navigate to the top-level administration context.

- 2. Choose the web access tab.
- 3. In Web Access Permissions, choose the name of the access group for which you want to view information. Review the following:
  - The list of features available for this group.
  - Whether the group is set as the default for the deployment. If it is, the group name will be followed by (Default). This is the group that all users of your deployment of Team Foundation Server will be assigned to by default.
  - Whether any users or groups are specifically assigned to this group. Unless you specifically configure users or groups to use this group, you will not see a list of identities assigned to this group.
- 4. To choose a different group to act as the default, choose that group in the explorer pane, and in Web Access Permissions for that group, choose Set as default web access.

If the group is already set as the default, this option will be unavailable.

For more information about what features are available for each access group, managing users in access groups, and auditing access, see <u>Change access levels</u>.

Home | Prepare for Installation | Install Prerequisites and Team Foundation Server | Configure Team Foundation Server to Support Your Development Teams | Create Back Up Schedule and Plan

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Create Back Up Schedule and Plan
Send Feedback

This fourth topic in the Single-Server Installation tutorial teaches you how to configure backups for a single-server deployment of Team Foundation Server (TFS) that also uses SharePoint Foundation 2010 and SQL Server Reporting Services. All of these products are installed on the same server.

#### ⊠Note

Before the release of <u>Cumulative Update 2</u>, there was a backup and restore power tool available for Team Foundation Server 2012. This tool is not compatible with the Scheduled Backups feature now available in this update. If you installed this tool before Cumulative Update 2, the update will uninstall the tool for you and attempt to import your settings into the Scheduled Backups feature. Do not install the power tool after installing Cumulative Update 2.

The examples in this tutorial topic follow Jill, an IT Administrator at the fictitious company, Fabrikam, Inc., as she installs and deploys Team Foundation Server in a single-server configuration in order to support her software development teams.

If you administer systems, you're probably already familiar with all the reasons why a good backup set is crucial. No one wants to be the administrator of a server that goes down without a complete set of backups in place. Fortunately, you can back up data for TFS by using the Scheduled Backups tool in the administration console. If you regularly back up those databases, you reduce the risk of losing productivity or data because of equipment failure or other unexpected events. Unlike previous methods, this tool also backs up the SharePoint databases that TFS uses, if TFS is configured to use SharePoint.

## **▲Important**

If you have not installed Cumulative Update 2, or your deployment has specific backup requirements not met by the Scheduled Backups features, you can manually configure backups for your deployment. For more information, see <u>Manually Back Up Team Foundation Server</u>.

In addition, if you are using the Enterprise or Datacenter edition of SQL Server and you want to restore the collection and configuration databases to a Standard edition of SQL Server then-before you make the backup-you must turn off SQL Server compression by following the steps in the <u>Microsoft Knowledge Base</u> <u>article</u>.

## **Required Permissions**

To perform the procedures in this tutorial, you must be a member of both of the following groups:

- The Administrators security group on the server that is running the administration console for Team Foundation.
- The SQL Server System Administrator security group. In addition, the service account for TFS (TFSService) must have SQL Server Perform Back Up and Create Maintenance Plan permissions set to Allow on each instance of SQL Server that hosts the databases that you want to back up, and Full Control on the network share, folder, or storage device where the backups will be kept.
- The Farm Administrators group in SharePoint Foundation 2010, or an account with the permissions required to back up the farm.

## **Create a backup plan**

1. Open the administration console for TFS and on the Scheduled Backups page, launch the wizard for creating a backup schedule.



If you don't see the Scheduled Backups node in the administration console, make sure that you have applied

#### Cumulative Update 2.

• Backups must be stored on a network-accessible location, and the service account for TFS must have Full Control access to that location. You can also choose how long a backup set will be kept, and the file extensions used for backup types.

Team Foundation Server	Scheduled Backups			
🦍 Schedu	led Backups Wiz	zard		
Details Alerts Schedules Review Readiness Checks	Provide Scheduled I You must specify a network specify has sufficient frees	Backups Ini k path for the b space to store b	tial Configuratior packup path. Be sure th packup files.	n Settings nat the folder you
Configure Complete	Network Backup Path:	\\fabrikamba	ackup\TFS\	Browse
	Backups Retention Days:	30	(j)	
	Advanced			
	Full Backup File Exten	sion:	bak	
	Transactional Backup	File Extension:	trn	

• If your server is configured with SMTP support, you can select email alerts for specific events. If not, all selections will be greyed out.

Team Foundation Server	Scheduled Backups
Schedul	ed Backups Wizard
Details Alerts Schedules Review Readiness Checks Configure Complete	<ul> <li>Provide Scheduled Backups Alert Settings</li> <li>Email alerts can notify TFS administrators in the event of a backup failure. TFS Notifications must be configured to enable this feature.</li> <li>Send email to TFS Administrators if a backup job fails</li> <li>Send email to TFS Administrators when a backup job succeeds</li> </ul>

• Choose between two default schedules, or create your own custom schedule.

Team Foundation Server S	cheduled Backups
Schedule	ed Backups Wizard
Details Alerts Schedules Review Readiness Checks Configure Complete	Select a Backup Schedule Nightly Full Backups () Manual Backups Only () Custom Schedule () Full Backup Schedule: 02:00 AM Sunday Friday Tuesday Saturday Wednesday Differential Backup Schedule: 02:00 AM Sunday Friday Sunday Friday Sunday Friday Sunday Friday Sunday Friday Sunday Friday Sunday Friday Sunday Friday Sunday Friday Monday Friday Sunday Friday Monday Friday

• Complete the wizard. If your deployment uses reporting, you will be prompted for a password in order to back up the encryption key for reporting.



• Once scheduled backups have been configured, you can allow them to run as scheduled. You can also choose to take an immediate backup, which will back up your data right away while leaving your scheduled plan in place. This is particularly recommended if your scheduled backups will not occur for a significant amount of time, or if you do not already have a recent backup available.

Home | Prepare for Installation | Install Prerequisites and Team Foundation Server | Configure Team Foundation Server to Support Your Development Teams | Create Back Up Schedule and Plan

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Restoring a Single-Server Deployment to New Hardware [Tutorial]
Send Feedback

No matter how well you maintain your hardware, server failures happen. As an experienced administrator, you know to prepare against such an event. If the server that hosts your deployment of Team Foundation Server (TFS) fails or is destroyed in a disaster, you can use the backups you made of the databases and of the SharePoint Foundation 2010 farm to restore the deployment to a new server. While restoring an entire deployment isn't simple, with reliable data backups and the installation media for the programs you need to install, you can restore your deployment to full functionality.

In addition to the backup data, you need the installation media for TFS, SQL Server 2008 R2 SP1, and SharePoint Foundation 2010, including access to all service packs and cumulative updates you applied to your original deployment. You must install and configure the software on the same operating system as the original server, and you should ensure that you restore the backups to the same versions, editions, and servicing levels of the software that you backed up.

This tutorial is designed as a companion tutorial to <u>Get Up and Running With a</u> <u>Single-Server Installation [Tutorial]</u>. It will be easier to follow if you're familiar with that tutorial. In addition, this tutorial assumes that you have installed <u>Cumulative Update 2</u> or later, which includes the Scheduled Backups feature.

## ⊠Note

This tutorial contains fictitious users, servers, and companies as part of the examples that help illustrate the procedures. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

## **Tutorial contents**

Title	Description
<u>Prepare the</u> <u>New</u> <u>Hardware</u>	Learn how to choose the right replacement hardware, review the prerequisites for a replacement server, and install the software you'll need in order to restore the deployment. As part of this process, you must choose a replacement server, and then install SQL Server 2008 R2 SP1, SharePoint Foundation 2010, and the tools that are required to reconfigure the restored TFS databases.
<u>Restore the</u> <u>Databases</u>	Learn how to restore the databases from backups, and to restore the SharePoint farm. As part of this process, you must restore the configuration, collection, and reporting databases, and restore the SharePoint farm from backup.

<u>Install and</u> <u>Configure</u> <u>Team</u> Foundation Server	Learn how to use the Application-Tier Only wizard to install the web services and applications for TFS. As part of this process, you must install Team Foundation Server and the extensions for SharePoint Products, and configure URLs, service accounts, reporting, and analysis resources to complete the data restoration part of the recovery.
Reconnect Services and Users	Learn how to complete configuration of TFS after its databases have been restored. As part of this process, you will start team project collections, add groups of users to the new server, and clear the data caches to help prevent data errors. Learn how to clear the work item cache on the new server, and learn how users can clear their version control caches on their own computers so that they don't experience data caching problems when connecting to the new server.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Prepare the New Hardware
Send Feedback

This topic, the first part of the Restore a Single-Server tutorial, teaches you how to install and configure the prerequisites, how to restore the databases and the SharePoint farm, and how to install but not configure Team Foundation Server (TFS) so that you can restore the databases on the new hardware.

The examples in this tutorial topic follow Jill, an IT Administrator at the fictitious company, Fabrikam, Inc., as she recovers from a hardware failure of the original deployment described in <u>Get Up and Running With a Single-Server</u> <u>Installation [Tutorial]</u>. The examples here will be easier to follow if you're familiar with that tutorial, especially the software installation procedures and the architecture described in <u>Install Prerequisites and Team Foundation Server</u>.

In this topic

- 1. Choosing hardware and naming the server
- 2. Installing SQL Server 2008 R2 on the new server
- 3. Installing SharePoint Foundation 2010 on the new server
- 4. Install Team Foundation Server

Required permissions

To perform the procedures in this topic, you must be a member of the Administrators security group on the server where you want to install the software.

# Choosing hardware and naming the server

Restoring a deployment that has unexpectedly gone offline due to a hardware failure is always stressful, and usually involves acquiring hardware as quickly as possible. You might be tempted to grab an unused machine and try to use it as the new server. However, for the long-term health of your deployment and to help ensure a successful restoration of your deployment, you should not use any server that has less RAM than the previous machine, and you should make sure that the new server is running the same operating system as the old server. You also must match the service pack level of the previous server as closely as possible.

In addition, this can be an opportune time to expand the hard drive space or processing power for your deployment, particularly if your previous server was not performing as quickly as you wanted, or if you were running low on hard drive space. Upgrading the numbers or speed of the processors, adding more RAM, or adding more hard drive space should not adversely affect your restoration, and you and your users can benefit from a faster, more powerful server.

## **Naming the server**

Your users will experience the least amount of confusion and inconvenience if you give the new server the exact same name as the old one. You will also find it much easier to restore the deployment if the new server has the same name as the old server. Unless you have compelling reasons not to reuse the name, such as underscores in the old server name that interfered with navigation in Team Web Access, use the same name for the new server as the old server.

## **The new server for the example deployment**

This example continues the tutorial that follows Jill, an IT Administrator at the fictitious company, Fabrikam, Inc., as she recovers from a hardware failure of the original deployment. Jill's previous server ran the 64-bit edition of the

Windows Server 2008 R2 operating system with Service Pack 1 installed, had a dual-core processor and10 GB of RAM, and had 300 GB of free disk space. When faced with the unexpected need for a new server, she briefly considers using another machine that exactly matches those specifications. However, she knows that the adoption of TFS is growing, and that several other departments are considering using her deployment. With that in mind, she negotiates funds to purchase a new server with a faster dual-core processor system, 16 GB of RAM, and two 500 GB hard drives. Just like the old server, this new server runs the 64-bit edition of Windows Server 2008 R2 with Service Pack 1 installed. She gives the new server the same name as the old server, FabrikamPrime.

# Installing SQL Server 2008 R2 on the new server

After you acquire a server that meets the requirements mentioned above and give it the same name, you must install the same version and edition of SQL Server on that new hardware. You must also make sure to choose the same collation settings as your previous deployment, or you will not be able to restore the databases you backed up for the old deployment.

## ĭTip

Most installations of SQL Server use the default collation settings. The default collation settings are determined by the Windows system locale on the server where you install SQL Server. The example deployment in the tutorial

Get Up and Running With a Single-Server Installation [Tutorial] used the default settings, and the new server has the same system locale as the old server, so the collation settings will remain the same by default.

In the example deployment, Jill installs SQL Server R2 using the exact same settings and choices she used when installing the software on the original server, and then applies Service Pack 1. Those choices are used in the procedure below, and are the same as those listed in <u>Install Prerequisites and Team Foundation</u> <u>Server</u>.

## To install SQL Server 2008 R2 to support Team Foundation Server

- 1. Launch the SQL Server Installation Center.
- 2. On the SQL Server Installation Center page, choose Installation, and then choose New installation or add features to an existing installation.
- 3. On the Setup Support Rules page, verify that all rules have passed, and then choose OK.

- 4. On the Product Key page, provide your product key, and then choose Next.
- 5. On the License Terms page, review the license agreement. If you accept the terms, select I accept the license terms. Optionally, you can select the check box to send usage data to Microsoft, and then choose Next.
- 6. On the Setup Support Files page, choose Install.
- 7. On the Setup Support Rules page, review the setup information. Correct any failure conditions, and then choose Next.
- 8. On the Setup Role page, choose SQL Server Feature Installation, and then choose Next.
- 9. On the Feature Selection page, select the following check boxes, and then choose Next:
  - Database Engine Services
  - Full-Text Search
  - Analysis Services
  - Reporting Services
  - Client Tools Connectivity
  - Management Tools Basic
  - Management Tools Complete

Features:	Description:
Instance Features	Includes the Database Engine, the core service for storing, processing and securing data. The Database Engine provides controled access and rapid transaction processing and also provides rich support for sustaining high availability. The Database Engine also provides support for the utility control point in the SQL Server Utility. Only Database Engine Services and Analysis Services can be clustered.

- 10. On the Installation Rules page, review any warnings and correct any failures, and then choose Next.
- 11. On the Instance Configuration page, choose Default instance, and then choose Next.
- 12. On the Disk Space Requirements page, review the information to make sure you have sufficient disk space, and then choose Next.
- 13. On the Server Configuration page, choose Use the same account for all SQL Server services. In the Use the same account for all SQL Server 2008 R2 services window, choose or specify **NT AUTHORITY\NETWORK SERVICE**, and then choose OK.

In the Startup Type column, specify Automatic for all services that you can edit, and then choose Next.

Service Accounts	Collation				
dicrosoft recommend	ls that you use a s	eperate account for each SQL S	erver service.		
Service		Account Name	Password	Startup Typ	e
SQL Server Agent		NT AUTHORITY\NETWO		Automatic	
SQL Server Database	e Engine	NT AUTHORITY\NETWO		Automatic	-
SQL Server Analysis	Services	NT AUTHORITY\NETWO		Automatic	-
SQL Server Reporting	g Services	NT AUTHORITY\NETWO		Automatic	-
SQL Full-text Filter D	aemon Launcher	NT AUTHORITY\LOCAL S		Manual	
SQL Server Browser		NT AUTHORITY\LOCAL S		Automatic	-

- 14. On the Database Engine Services page, on the Account Provisioning tab, choose Windows authentication mode and then choose Add Current User to add your account as an administrator for this instance of SQL Server. Optionally add any other user accounts for users you want to act as database administrators, and then choose Next.
- 15. On the Analysis Services Configuration page, on the Account Provisioning tab, choose Add Current User to add your account as an administrator for the analysis services database. Optionally add any other user accounts for users you want to act as administrators, and then choose Next.
- 16. On the Reporting Services Configuration page, choose Install the native mode default configuration, and then choose Next.
- 17. On the Error Reporting page, choose whether to send information about errors to Microsoft, and then choose Next.
- 18. On the Installation Rules page, review any failures or warnings, and then choose Next.
- 19. On the Ready to Install page, review the list of components to be installed, and if they match the list of features shown in the illustration below, then choose Install. If you need to make any changes, choose Back.



On the Installation Progress page, optionally monitor the installation progress of each component. When all components have finished installing, the Complete page appears. Review any messages, and then close the page.

20. After you finish the installation, visit Microsoft Update to apply the same service packs or updates that you applied to the installation on the old server. TFS requires SQL Server 2008 R2 to have Service Pack 1 applied, so make sure that you install that service pack. Do not apply any service packs or updates that were not applied to that server.

# Installing SharePoint Foundation 2010 on the new server

Unlike a new installation of Team Foundation Server, you cannot use the installation wizard for TFS to install SharePoint Foundation 2010 for you. If you want to be able to restore the team project portals and other information used in the SharePoint Foundation 2010 portion of your deployment, you must first install SharePoint Foundation 2010 manually, and then restore the farm.

## **Using SharePoint Tools to install SharePoint Foundation 2010**

You can use the tools and features provided with SharePoint Foundation 2010 to install it on the new server. You should choose the same deployment configuration as your previous deployment. For example, if you installed SharePoint Foundation 2010 automatically during the installation process for Team Foundation Server in your original deployment, you should choose to install SharePoint Foundation 2010 on a single server, but using the installation of SQL Server you have already installed on the new server. You should also run the Microsoft SharePoint Products Preparation Tool prior to installing SharePoint Foundation 2010.

## To install SharePoint Foundation 2010 in a single-server farm configuration

- 1. Launch the installation media for SharePoint Foundation 2010. On the SharePoint Foundation 2010 Start page, choose Install SharePoint Foundation.
- 2. On the Read the Microsoft Software License Terms page, review the terms, select the I accept the terms of this agreement check box, and then choose Continue.
- 3. On the Choose the installation you want page, choose Server farm.
- 4. On the Server Type tab, choose Complete.

If you want to install SharePoint Foundation 2010 in a custom location, choose the Data Location tab, and then either type the location or choose Browse to specify the location.

- 5. Choose Install Now.
- 6. When setup completes, choose Close. Do not configure the initial farm.

## **■Using Windows PowerShell to install SharePoint Foundation** 2010

While there are a number of ways to install SharePoint Foundation 2010, including the tools provided with SharePoint Foundation 2010, restoring the farm requires Windows PowerShell. You can also use Windows PowerShell to install SharePoint Foundation 2010. For convenience, consider using Windows PowerShell for both installing and restoring SharePoint Foundation 2010 on the new server. For more information about Windows PowerShell, see

Getting Started: Windows Server 2008 R2 Administration with Windows PowerShell.

## To install SharePoint Foundation 2010 using Windows PowerShell

- 1. Open a Windows PowerShell command prompt.
- 2. Input the following command, where Drive is the location of the installation media for SharePoint Foundation 2010:

## Copy Code

Install-SharePoint -SetupExePath "Drive:\SharePoint 2010\Setup\:

This installs SharePoint Foundation 2010 using a PID key in a farm deployment, but does not configure it or create any databases. Instead, you will restore the farm and its databases to this installation.

ĭTip

As an alternative, you can choose to use a configuration XML file with the Install-SharePoint command to install SharePoint Foundation 2010.

For more information, see <u>Install SharePoint Foundation 2010 by using</u> <u>Windows PowerShell</u>.

## **Install Team Foundation Server**

To restore the databases using the Scheduled Backups features, you must install, but not configure, the Team Foundation Server software.

## **To install Team Foundation Server binaries and tools**

- 1. Launch the installation media for Team Foundation Server. On the Team Foundation Server Setup page, choose Install.
- 2. When the installation completes, the Team Foundation Server Configuration Center opens. Choose Cancel.

.erver 2012 Start Wizard Close
<ul> <li>You have not already installed Team Foundation Server.</li> <li>You have not set up load balancing infrastructure.</li> </ul>
You might not want to use the wizard if one or more of the following statements are true:
<ul> <li>You want to configure high availability for Team Foundation Server in a load-balanced environment.</li> <li>You want to move Team Foundation Server from one server to another.</li> <li>You want to restore Team Foundation Server (Disaster Recovery).</li> </ul>
You might want to use the wizard if one or more of the following statements are true:
Supported on Windows Server and Client operating systems
Use the Application-Tier wizard to add another Team Foundation Server to an existing deployment after the initial installation has been completed.
About this Wizard

The administration console opens automatically in an unconfigured state. This is expected.

3. If you haven't installed

<u>Cumulative Update 2</u> or later, install it now.

Home | <u>Prepare the New Hardware</u> | <u>Restore the Databases</u> | <u>Install and</u> <u>Configure Team Foundation Server</u> | <u>Reconnect Services and Users</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Restore the Databases
Send Feedback

This topic, the second part of the Restore a Single-Server tutorial, teaches you how to use the backups you made of the original server to restore the databases for Team Foundation Server (TFS) and the SharePoint farm on the new server.

You can use the backups you made of the original deployment to restore the data from that deployment to the new server. When restoring the data, be sure to restore all databases and the SharePoint Farm to the same point in time. If you followed the guidance in <u>Create Back Up Schedule and Plan</u>, you used the Scheduled Backups feature to create your backups. You will use these backups to restore your data.

If you manually configured your backups, you cannot use the Restore wizard in Scheduled Backups to restore those databases. You must manually restore them using the software you used to back them up.

## **▲Important**

To follow these procedures, you must have installed <u>Cumulative Update 2</u> for TFS 2012. In addition, if your original deployment used the Enterprise or Datacenter editions of SQL Server, and you want to restore databases to a server running Standard edition, you must use a backup set that was made with SQL Server compression disabled. Unless you disable data compression, you will not be able to successfully restore Enterprise or Datacenter edition databases to a server running Standard edition. To turn off compression, follow the steps in the <u>Microsoft Knowledge Base article</u>.

The examples in this tutorial topic follow Jill, an IT Administrator at the fictitious company, Fabrikam, Inc., as she recovers from a hardware failure of

the original deployment described in <u>Get Up and Running With a Single-Server</u> <u>Installation [Tutorial]</u>. The examples here will be easier to follow if you're familiar with that tutorial.

In this topic

- <u>Restore Team Foundation Server databases</u>
- Restore the SharePoint farm

**Required** permissions

To perform these procedures, you must be a member of the following groups or have the following permissions:

- A member of the Administrators security group on the server.
- Either a member of the SQL Server System Administrator security group, or your SQL Server Perform Back Up and Create Maintenance Plan permission must be set to Allow.
- A member of the sysadmin security group for the databases for Team Foundation.
- A member of the Farm Administrators group.

## ■Restore Team Foundation Server databases

Installing and configuring software isn't sufficient to recover a deployment. You must restore the data before your users will be able to get back to work. There's a wizard to help you do this..

## To restore databases

1. To start the Restore wizard, open the administration console for TFS and navigate to Scheduled Backups.



2. Specify the path to the backup set and choose the set you want to use for restoration.

Team Foundation Server Restore Databases Wizard						
🙀 Restore [	Databases Wiza	ard				
Backup Sets Databases Review Readiness Checks Pestore Databases	Select Backup Set Point to the backup pat Server to. Note: The mo	S h and select the point in ti st recent backup has been	me you want to restore your selected by default.	т		
Complete	Network Backup Path:	\\fabrikambackup\TFS\	Browse			
	Date 3/14/2013 5:19:55 PM 3/14/2013 4:49:54 PM 3/14/2013 4:19:54 PM 3/14/2013 3:49:54 PM 3/14/2013 3:24:08 PM 3/14/2013 3:19:57 PM 3/14/2013 2:19:53 PM 3/14/2013 2:19:53 PM 3/14/2013 1:49:54 PM 3/14/2013 1:29:28 PM 3/14/2013 1:20:52 PM		Size (MB) 192 192 191 191 190 100 100 180 179 179 38 178			

3. Complete the wizard and restore the databases.

Team Foundation Server Restore Databases Wizard				
🌆 Restore Da	atabases Wizard			
Backup Sets Databases Review	Configuration Progress			
Readiness Checks Restore Databases				
Complete	<ul> <li>Granting Required Permissions for Restore Operation</li> <li>Restoring Databases</li> <li>Saving Settings</li> </ul>	Complete Complete Complete		
	Configuration completed successfully. Click Nex	t to view deta		

In the example deployment, Jill uses the Restore wizard to restore the following databases:

- TFS\_Warehouse
- TFS\_DefaultCollection

This is the default name of the collection database. If you customized the name, make sure to use that name.

- TFS\_Configuration
- ReportServer
- ReportServerTempDB
- WSS\_Config
- WSS\_AdminContent
- WSS\_Content

The Restore wizard also restores the encryption key for SQL Server Reporting Services as part of its operation. You can choose to restore that key manually, but it should not be necessary.
### **Restore the SharePoint farm**

Although the Restore wizard restored the SharePoint databases used by your deployment, it cannot restore the farm. You must use Windows PowerShell with the Restore-SPFarm command to restore the backup you made of the SharePoint Farm. In some cases, you can choose to use the Central Administration website to restore the farm instead of Restore-SPFarm, but the PowerShell command is the preferred method for restoring a farm.

To restore a farm, either you must be logged in with an account that is a member of the Farm Administrators group, or you must provide the credentials for an account that is a member of that group when prompted to do so.

## To restore the farm for SharePoint Foundation 2010 using Restore-SPFarm

- 1. Open Windows PowerShell or SharePoint 2010 Management Shell.
- 2. At the Windows PowerShell command prompt, enter the following command, where *UNCPath* is the fully-qualified UNC path of the directory where the farm backup is located:

Restore-SPFarm -Directory UNCPath -RestoreMethod Overwrite

This command will restore the farm using the most recent backup available. If you want to use a different backup, you must specify which backup to restore by using the -BackupID parameter with the GUID of the specific backup you want to use.

3. At the Windows PowerShell command prompt, enter the following command, where *ServiceApplicationID* is the GUID of the restored farm:

Start-SPServiceInstance -Identity ServiceApplicationID

### ĭTip

If you do not know the GUID, you can use the Get-SPServiceInstance

command to list the service instance GUIDs for all SharePoint applications on the server.

4. For more information about restoring a farm, see

Restore a farm (SharePoint Foundation 2010) and Restore-SPFarm.

### To restore the farm using SharePoint Central Administration

1. Open SharePoint Central Administration, and on the Home page, in the Backup and Restore section, choose Restore from a backup.

The Restore Wizard opens.

- 2. On the Restore from Backup Step 1 of 3 page, choose the farm backup job that you want to restore, and then choose Next.
- 3. On the Restore from Backup Step 2 of 3 page, select the check box next to the farm option, and then choose Next.
- 4. On the Restore from Backup Step 3 of 3 page, in the Restore Component section, make sure that Farm appears in the Restore the following component list. In the Restore Only Configuration Settings section, choose Restore content and configuration settings. In the Restore Options section, under Type of Restore, choose Same configuration. When a dialog box appears asking you to confirm your choices, choose OK, and then choose Start Restore.
- 5. Monitor the general status of the recovery as it appears in the Readiness section of the Backup and Restore Job Status page. The status automatically updates every 30 seconds. You can also choose to manually update the status by choosing Refresh.
- 6. When restoration is complete, return to the Home page in Central Administration. In Application Management, choose Manage services on server.
- 7. On the Services on Server page, find the service applications for the restored farm, and in the Actions column, choose Start for each of those

service applications.

8. For more information about restoring a farm, see <u>Restore a farm</u> (SharePoint Foundation 2010).

Home | <u>Prepare the New Hardware</u> | <u>Restore the Databases</u> | <u>Install and</u> <u>Configure Team Foundation Server</u> | <u>Reconnect Services and Users</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Install and Configure Team Foundation Server
Send Feedback

This topic, the third part of the Restore a Single-Server tutorial, teaches you how to install Team Foundation Server (TFS) using the Application-Tier Only wizard. You will also learn how to install the Extensions for SharePoint Products on the new server. Additionally, you will learn how to configure SharePoint Foundation 2010, reporting, and analysis services to work with the new server.

The examples in this tutorial topic follow Jill, an IT Administrator at the fictitious company, Fabrikam, Inc., as she recovers from a hardware failure of the original deployment described in <u>Get Up and Running With a Single-Server</u> <u>Installation [Tutorial]</u>. The examples here will be easier to follow if you're familiar with that tutorial.

In this topic

- 1. Install Team Foundation Server and the Extensions for SharePoint Products
- 2. Configure SharePoint Foundation 2010
- 3. Configure reporting and Analysis Services

#### **Required** permissions

To perform this procedure, you must be a member of the following groups or have the following permissions:

- A member of the Administrators security group on the server.
- A member of the Team Foundation Administrators group.
- Either a member of the SQL Server System Administrator security group,

or your SQL Server Perform Back Up and Create Maintenance Plan permission must be set to Allow.

- A member of the sysadmin security group for the databases for Team Foundation.
- A member of the Farm Administrators group.

### ■Install Team Foundation Server and the Extensions for SharePoint Products

By restoring the databases that TFS uses, you have already restored the data tier of the original deployment. Now you must complete the restoration process by installing the application tier-the application and services that compose Team Foundation Server-by using the Application-Tier Only wizard. You must also install the extensions used by TFS and SharePoint Foundation 2010 by running the Extensions for SharePoint Products wizard.

### To install the application tier and the extensions



1. Open the administration console and start the configuration process.

If you don't see the administration console for TFS, you haven't installed the TFS software on the server. To learn how, go to Install Team Foundation Server. 2. Launch the application-tier only wizard.



3. Specify the name of the SQL Server where you just restored the databases and choose List Available Databases to populate the list. Choose the configuration database.

In this particular example, Jill specifies FKPMALTERNATE\SQLExpress in SQL Server Instance because she is using a named instance (SQLExpress) on the new server, and highlights TFS\_Configuration. In the ideal case, the new SQL Server instance would have the same name as the prior server (FABRIKAMPRIME).

Team Foundation Server Appl	ication-Tier Only Wizard		
🌆 Applicatio	n-Tier Only W	/izard	
Welcome Databases Application Tier Settings	Specify a Team F Type in a SQL Server in	Foundation Server configuration databa Instance path, click List Available Databases, then select	
	SQL Server Instance:		
Review Readiness Checks	FABRIKAMPRIME\SqlExpress		
Configure Complete	onfigure Databases:		
	Database Name	Version	
	Tfs Configuration	Microsoft Team Foundation Server 2012 Update 2	

4. Provide the same service account information as used in the prior deployment, if possible.

In the example deployment, Jill makes sure that NT AUTHORITY\NETWORK SERVICE appears in Service Account, and that the authentication method is set to the default, NTLM.

ion-Tier Only Wizar	d	
Only Wiza	ard	
Provide the se	ttings for your new Application-Tier	
Provide the service	account to use for your new Application-Tier.	
Server Url:	http://FABRIKAMPRIME:8080/tfs	
Service Account:	NT AUTHORITY\NETWORK SERVICE	<b>(i)</b>
Password:		Test
Authentication Met	hod	
NTLM		
C Negotiate (Kerb	eros)	
Negotiate (Kerb but may require	eros) is the more secure configuration to use with Integrated Windows aut additional configuration by the domain administrator. <u>More information.</u>	nentication,

5. Complete the wizard.



- 6. In the Configuration Center, choose Configure Extensions for SharePoint Products and complete that wizard to install the extensions.
- 7. Open the administration console, and in the Application Tier pane, update or refresh the URLs to reflect the new URLs for the restored server. Even if you have not changed the name or ports used for the restored server, you should reapply this information.



Notification U The Notific the text of	e-mail alerts.	em and is the URL used in
http://FA	RIKAMPRIME:8080/tfs	🗸 Iest
Server URL		
Server URL The Server should be http://go.n	URL is used for server-to-server communication. Under most confi et to the localhost. More information about the Server URL can be icrosoft.com/fwlink/?LinkID=155633.	gurations, the Server URL found at
Server URL The Server should be http://go.n	URL is used for server-to-server communication. Under most confi set to the localhost. More information about the Server URL can be icrosoft.com/fwlink/?LinkID=155633. alhost	igurations, the Server URL e found at

8. After finishing the update, verify that the URLS appear correctly.

Team Foundation Server Administration Const	sole	
File Help		
FABRIKAMPRIME     FABRIKAMPRIME     Fabrication Tier     Fabrication Tier     Fabrications     Fabrications     Fabrications     Fabrications     Fabrications     Fabrications     Fabrications     Fabrication	Application     Application Tier Sum     Service Account:     Web Site:     Application Rool:	Tier mary NT AUTHORITY\NETWORK SER Team Foundation Server Microsoft Team Foundation Se
<ul> <li>Extensions for SharePoint Products</li> <li>Scheduled Backups</li> <li>Logs</li> <li>Additional Tools and Components</li> <li>Visual SourceSafe Upgrade</li> <li>PreEmptive Analytics</li> </ul>	Application Pool: Authentication: Notification URL: Server URL: Web Access URL: Machine Name: Ports: Virtual Directory: Version:	NTLM http://fabrikamprime:8080/tfs http://fabrikamprime:8080/tfs http://fabrikamprime:8080/tfs FABRIKAMPRIME 8080 /tfs 11.0.60315.0 (Tfs2012.Update3

9. Reapply the service account for TFS.

happlication	n Tier		😰 Refresh
Application Tier Summa	ary		
Service Account: Web Site: Application Pool:	NT AUTHORITY\NETWORK SERVICE Team Foundation Server Microsoft Team Foundation Server Application Pool	 	Update Password Change Account Reapply Account
Reapply Service Account Select a system or domain a Team Foundation Server app	ccount to use as the service account for the vication tier account.		Authentication Settings Group Membership Administer Security
Use a system account:     Use a user account:     Account Name:     Password:	NT AUTHORITY\NETWORK SERVICE	<u>Iest</u>	Change URLs
Fassword:	ОК Салс	el	

### **Configure SharePoint Products**

After you install the application tier and the extensions, you must reconfigure the settings for the SharePoint web applications that your deployment uses.

### To configure the settings for SharePoint web applications

- 1. Open the administration console for Team Foundation.
- 2. In the navigation bar, choose SharePoint Web Applications.
- 3. In the SharePoint Web Applications list, highlight the web application used by the original deployment, and then choose Change.

#### ĭTip

By default, the web application should appear automatically in the list of web applications. If it does not, choose Add, and then add it manually.

- 4. The SharePoint Web Application Settings window opens. In Web Application URL and Central Administration URL, review the settings, and if necessary, specify or change the values to reflect the URLs on the new server.
- 5. When you are satisfied with the settings, choose OK.

For more information about how to configure SharePoint web applications as part of your deployment, see

<u>Add a SharePoint Web Application to Your Deployment</u> and Interactions Between SharePoint Products and Team Foundation Server.

### ■Configure reporting and Analysis Services

After you have configured SharePoint Foundation 2010, you must redirect Team Foundation Server to the location of the report server, restart the warehouse, and manually rebuild the database for Analysis Services.

#### ⊠Note

You must complete this procedure even if you restored the TFS\_Warehouse and TFS\_Analysis databases, as the previous section describes.

### To reconfigure reporting and Analysis Services

- 1. In the navigation bar for the administration console, choose Reporting.
- 2. On the Reporting page, choose Edit.
- 3. In the Take Offline dialog box, choose OK.

The Reporting dialog box opens.

- 4. Select the Use Reporting check box.
- 5. On the Warehouse tab, in Server, specify the name of the report server, and in Database, type the name of the warehouse database. Optionally choose Test Connection to verify that the database is valid.
- 6. On the Analysis Services tab, in the Server list, specify the name of the server that is running SQL Server Analysis Services. In Database, specify the name of the analysis services database, and in Account for accessing data sources, specify the username and password of the data sources account.
- 7. On the Reports tab, in the Server list, specify the name of the report server,

and then choose Populate URLs. In Username and Password, specify the account name and password (if any) for the data sources account. In Default Path, specify the relative path for the location where reports are stored, and then choose OK.

- 8. In the administration console, choose Start Jobs to restart reporting.
- 9. Open a Command Prompt window, and then change directories to %ProgramFiles%\Microsoft Team Foundation Server 11.0\Tools.
- 10. Enter the following command to rebuild the database for Analysis Services:

TFSConfig RebuildWarehouse /AnalysisServices /ReportingDataSourcePassword:Password

Password is the password for the data sources account for Reporting Services.

In the example deployment, Jill types the following at the command line: **TFSConfig RebuildWarehouse /AnalysisServices** /**ReportingDataSourcePassword:**password.

11. Wait until the command is successfully completed.

Home |

<u>Prepare the New Hardware | Restore the Databases | Install and Configure Team</u> <u>Foundation Server | Reconnect Services and Users</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Reconnect Services and Users
Send Feedback

This topic, the fourth part of the Restore a Single-Server tutorial, teaches you how to start team project collections and verify user groups on the new server. You will also learn how to clear the data caches on the new server, and how to help ensure that your users do not experience data caching problems when they connect to the new server.

The examples in this tutorial topic follow Jill, an IT Administrator at the fictitious company, Fabrikam, Inc., as she recovers from a hardware failure of the original deployment described in <u>Get Up and Running With a Single-Server</u> <u>Installation [Tutorial]</u>. The examples will be easier to follow if you're familiar with that tutorial.

In this topic

- 1. Start team project collections and verify permissions
- 2. <u>Refresh the version control cache on client computers and reconnect to</u> <u>Team Foundation Server</u>

#### **Required permissions**

To invoke the StampWorkitemCache web method, you must be a member of the Administrators security group on the server.

To use the **tf workspaces** command on the client computer, your **Read** permission must be set to **Allow**.

# ■Start team project collections and verify permissions

After you have completed restoring the data and software on the new server, start the team project collections, and then make sure that all users, groups, and service accounts for your deployment are configured with the permissions that they require to perform tasks and function correctly.

You cannot automatically migrate all permissions. For example, administrators for Team Foundation must be members of the local Administrators group on the server so that they can open the administration console, and you must manually add them to that group. If you followed the tutorial on

<u>Get Up and Running With a Single-Server Installation [Tutorial]</u>, you can use the user groups you created in <u>Prepare for Installation</u> to make this task easier.

### To start a team project collection

1. Open the administration console, and in Team Project Collections, start the default collection.



2. Repeat this action for any other team project collections in the deployment.

### To verify permissions

• Log on to the server and make sure that users, groups, and service accounts are configured with the permissions that they need to operate.

For example, in the example deployment, Jill adds the group "TFS\_ProjectContributors"-which resolves to the friendly name Fabrikam Developers and Testers-at the appropriate permissions level for users who will contribute fully to the project in Team Foundation Server, SQL Server Reporting Services, and SharePoint Foundation 2010. For examples of how she does this, see <u>Configure Team Foundation Server to Support Your</u> <u>Development Teams</u>.

For more information, see <u>Add Users to Team Projects</u>, <u>Set Administrator</u> <u>Permissions for Team Project Collections</u>, <u>Set Administrator Permissions</u> <u>for Team Foundation Server</u>, and <u>Service Accounts and Dependencies in</u> <u>Team Foundation Server</u>.

### **Refresh the work item cache**

#### ⊠Note

This procedure is optional. You should perform it only if you receive errors with work item tracking.

To update the cache for tracking work items, you invoke the StampWorkitemCache web method. This method forces client computers to update the cache the next time that they connect to the application-tier server. This method also synchronizes the workspaces that are defined on the client computers.

### To refresh the cache for tracking work items on client computers

- 1. On the new server, open Internet Explorer.
- 2. In the Address bar, enter the following address to connect to the ClientService web service:

http://PublicURL:8080/VirtualDirectory/WorkItemTracking/v3.0/ClientS

#### ⊠Note

Even if you are logged on with administrative credentials, you might need to start Internet Explorer as an administrator, and you might be prompted for your credentials.

3. Choose StampWorkitemCache, and then choose Invoke.

#### ⊠Note

The StampWorkitemCache method returns no data.

### Refresh the version control cache on client computers and reconnect to Team Foundation Server

To prevent workspace errors from occurring during version control or build operations in Team Foundation, you have to update the data cache on client computers.

To refresh the version control cache, each user must run the **tf workspaces** command on any computer that uses version control and that connects to the restored server.

### To refresh the version control cache on client computers

- 1. On the client computer, open a Command Prompt window with administrative permissions, and change directories to Drive:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\IDE.
- 2. At the command prompt, enter the following command, including the URL of the collection, which includes the server name and the port number of the new server:

tf workspaces
/collection:http://ServerName:Port/VirtualDirectoryName/CollectionName

In the example deployment, one of Jill's developers needs to refresh the version control cache for a project that is a member of the DefaultCollection collection, which is hosted in the FabrikamPrime deployment of Team Foundation Server. He types the following string:

tf workspaces /collection:http://FabrikamPrime:8080/tfs/DefaultCollection

For more information, see Workspaces Command.

Home |

Prepare the New Hardware | Restore the Databases | Install and Configure Team Foundation Server | Reconnect Services and Users Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Add reports to a team project
Send Feedback

By adding a report server to your TFS (on-premises) deployment, you can access a wealth of data about your team's projects, such as build quality, bug trends, burndown, and test progress. SQL Server Reporting Services (SSRS) reports provide insight to help teams manage work and improve processes.



After completing the sequence of tasks, you'll be able to access the default reports provided with the process template used to create your team project.



Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Add a report server
Send Feedback

On-premises installations of TFS can include reports to help you manage your software development projects. However, you need a report server as part of your deployment in order to use them. If you don't have one, you can add SQL Server Reporting Services to your deployment.

### ĭTip

If you're using Team Foundation Service (http://tfs.visualstudio.com/), adding a report server isn't an option for you at this time. Additionally, if you installed TFS on a client operating system, such as Windows 7, you can't add reporting as described here, because it isn't supported on client operating systems.



### What do I need to know before adding a report server?

Q: What kind of report server can I add?

A: You'll need to add SQL Server Reporting Services to provide a report server for TFS. This can be added to the SQL Server instance that is supporting TFS, or to a different instance.

Q: What permissions do I need?

A: You need to be a member of the local administrators group on the server, the Team Foundation Administrators group in TFS, and the sysadmin group in SQL Server, or have the equivalent permissions. If you don't have them, <u>get those</u> <u>permissions now</u>.

Q: Can I add reporting if TFS is installed on a client operating system?

A: If it's a client operating system, you can't add reporting and analysis services directly. Your choices aren't easy. If you still want to add reporting, either <u>move</u> the team project collection to a deployment of TFS that supports reporting, or <u>restore your entire deployment to a different server</u> that is running a server operating system and a version of SQL Server that supports reporting.

Q: How do I find out if I already have reporting available for TFS?

A: Look in the administration console for Team Foundation Server. If you see a reporting node and it shows that reporting is configured and enabled, it's already there.



Jump to the bottom of this topic to confirm that reports are available for the team project collection, and then move on to <u>Upload reports to a team project</u>.

Q: Do I need to back up my data first?

A: You should have a recent backup available. If you don't, you can <u>install</u> <u>Cumulative Update 2</u>, and make a backup using Scheduled Backups.

# ■Verify your options and upgrade if necessary

- 1. Find out what version and edition of SQL Server you have <u>by following the instructions here</u>. If the answer is SQL Server Express, keep following these steps. If it's any other edition, but you don't see any version and edition information about reporting or analysis services, jump ahead to <u>Add</u> <u>Reporting Services</u>. If you do see information about these services, then your SQL Server already has them installed, and you can jump ahead to <u>Add Reporting to TFS</u>.
- 2. If your deployment is using SQL Server Express, <u>upgrade to a different</u> <u>edition</u>. For more information about the editions that support reporting in TFS, <u>go here</u>.

### Add Reporting Services and Analysis Services to SQL Server

After you confirm that you have an edition of SQL Server that supports reporting (or you upgrade to one), add the reporting and analysis services features if they aren't already configured.

1. Open the SQL Server Installation Center to add features to an existing installation.



You might have to restart your computer during this process, particularly if you've just finished upgrading from SQL Server Express.

2. Once your server has passed all the setup rules checks, pick the instance to add features to. Tip: If you upgraded from SQL Server Express, you'll have a named instance called SQLEXPRESS.

Installation Type Perform a new installation or add	d features to an existir	ng instance of SQL Serve	er 2012.	
Setup Support Rules Installation Type Feature Selection Installation Rules Disk Space Requirements Error Reporting Installation Configuration Rules Ready to Install Installation Progress Complete	<ul> <li>Perform a new Select this opt components s</li> <li>Add features t</li> <li>SQLEXPRESS Select this opt want to add th within an insta</li> <li>Installed instances</li> </ul>	v installation of SQL Ser tion if you want to insta uch as SQL Server Mana to an existing instance of tion if you want to add f the Analysis Services feat ance must be the same of s:	ver 2012 Il a new instance of SC agement Studio or Inte f SQL Server 2012 eatures to an existing sures to the instance to edition.	)L Serv egratio instan hat con
	Instance Name SQLEXPRESS	Instance ID MSSQL11.SQLEXPR	Features SQLEngine,SQLEn	Edition Enterp

- 3. In Feature Selection, add these features:
  - Analysis Services
  - Reporting Services Native
  - Management Tools Complete (not pictured)

### **Feature Selection**

Select the Enterprise features to install.

Setup Support Rules	Features:		Fe
Installation Type	Instance Features		
Feature Selection	Database Engine Services		
Installation Rules	SQL Server Replication	=	
Disk Space Requirements	Full-Text and Semantic Extractions for Searc		
Server Configuration	Data Quality Services		
Analysis Services Configuration	<ul> <li>Analysis Services</li> <li>Reporting Services - Native</li> </ul>		Pr
Reporting Services Configura	Shared Features		
Error Reporting	Reporting Services - SharePoint		
Installation Configuration Rules	Reporting Services Add-in for SharePoint Prod		

4. Accept the defaults for service accounts in Server Configuration, unless your deployment has specific service account needs. Make sure all services are set to start automatically.

Service Accounts	Collation				
Microsoft recomme	nds that you u	se a separate account for each SQ	L Server servi	ce.	
Service		Account Name	Password	Startup Type	
COL Conver Analysi	is Services	NT Service\MSOLAP\$SQ		Automatic	
SQL Server Analys				Automatic	-
SQL Server Analysi	ing Services	NT Service\ReportingServer		Automatic	1

- 5. Add your user account in Analysis Services Configuration.
- 6. Choose Install and Configure for reporting services if you can; if not, choose Install Only. If you're adding features after upgrading from SQL Server Express, Install Only might be the only option available.



7. Finish the wizard, including any server restarts that might be required.

If you had to choose Install Only when you added reporting, you'll need to open Reporting Services Configuration Manager and do some additional configuration. If not, jump ahead to

Add Reporting to TFS.

### **Configure Reporting Services Manually**

1. Open Reporting Services Configuration Manager and connect to your report server.

	Reporting Services Configuration Connection	×
SQL Serv Reporting Se	rvices	
Please specify a server na instance to configure.	ame, click the Find button, and select a report server	
Server Name:	FABRIKAMPRIME Find	
Authentication Type:	SQLEXPRESS V	
0	Connect Cance	1

- 2. Once you've connected, make sure the service is started, then go to the Web Service URL page and apply all the defaults.
- 3. Now you'll need to create a database for reporting. On the Report Server Database page, the Change Database option will open the Report Server Database Configuration Wizard assist you in creating that database.



4. When you specify the database server connection, include the instance name as well as the server name, separated by a slash (\).

	Report Server Databa	ase Configuration Wiz
Change Databas Choose whether to crea	e ate or configure a report server data	ibase.
Action		
Database Server	choose a local or remote instance credentials that have permission	te of a SQL Server Database Engine and specify to connect to that server.
Database		
Credentials	Connect to the Database Server:	
Summary		
Progress and Finish	Server Name:	FABRIKAMPRIME\SQLEXPRESS
	Authentication Type:	Current User - Integrated Security
	Username:	NORTHAMERICA\ctsoita
	Password:	
		Test Connection

If you aren't using a named instance, you can just provide the server name.

5. Accept the default values on the rest of the pages of the wizard and wait for it to finish. This can take a few minutes.

6. Accept all the defaults on the Report Manager URL page.

Reporting Services Configura	ation Manager	
	Report Manager URI	
Service Account	Configure a on the URL.	URL to access Report Manger. Click Advan
<ul> <li>Database</li> <li>Report Manager URL</li> <li>E-mail Settings</li> </ul>	The Report Ma the is provided Report Manager Site	anager virtual directory name is not config d, and then click Apply. Identification
🔼 Execution Account	Virtual Directory:	Reports_SQLEXPRESS
Recryption Keys	URLs:	http://FABRIKAMPRIME:80/Reports_SQLE

You've now fully configured reporting on the SQL Server instance that supports TFS. It's time to add the reporting features to TFS!

## **Add Reporting to TFS**

You'll need to add reporting in two places: to TFS itself, and then to your team project collection.

1. Open the administration console for TFS and go to the Reporting node to start the configuration process.



2. Select the Use Reporting check box. On the Warehouse tab, provide the name of the report server and instance for the warehouse database. Use the default name for the database (TFS\_Warehouse), and test the connection.

File Help	Reporting	x
FABRIKAMPRIME     FABRIKAMPRIME     Application	✓ Use Reporting	
Team P	Warehouse Analysis Services Reports	
Reporting Lab Man Proxy Server Build Confi Extensions Logs	Server: FABRIKAMPRIME  Specify non-default instance:  SQLEXPRESS  Database:	
	Tfs_Warehouse	
	Test Connection	
	The database that you specified does not exist. A new database will be create and populated for use with Team Foundation Server.	

Tip: Because you're configuring reporting for the first time, the databases don't exist yet. They will be created when you finish the configuration process.

3. Provide the same server and instance information for Analysis Services, but use the default name for the analysis database (TFS\_Analysis). Provide an account name and password for a data sources account, a special account you've created just for this purpose. (You can read more about these accounts

here.)

File Help	Reporting	×
FABRIKAMPRIME	✓ Use Reporting	
Gan P SharePo Reporting	Warehouse Analysis Services Reports Server:	]
Lab Man	FABRIKAMPRIME	
Build Confi Extensions	Specify non-default instance:	
	SQLEXPRESS T	
	Tfs_Analysis	
	Test Connection	
	Account for accessing data sources	
	Username:	
	Password:	
	The database that you specified does not exist. A new database will be create and populated for use with Team Foundation Server.	

Tip: Test the connection before continuing.

• For Reports, provide the server name and the instance name in Server, separated by a slash. Use Populate URLs to automatically populate the next two boxes. Add the same data sources account information you provided on the previous tab.

File Help	Reporting		
<ul> <li>FABRIKAMPRIME</li> <li>FABRIKAMPRIME</li> <li>Application</li> </ul>	✓ Use Reporting		
🏹 Team P	Warehouse Analysis Services Reports		
Reporting Lab Man Proxy Server Build Confi Extensions Logs	Server:		
	FABRIKAMPRIME\SQLE	XPRESS 💌	Populate URLs
	URLs for Report Ser	ver	Prove d Comment
	Report Manager:	http://FABRIKAMPRIME:80/	Report_SQL
	Account for accessir	ng data sources	
	Username:		
	Password:		
	Team Foundation Se	erver Reports	
	Default Path: /Tfsl	Reports	

Once you finish, databases will be created, and the warehouse will start building.


• Now add the report server to the team project collection. Edit the information in Reports Folder.



#### ĭTip

If you don't see the Reports Folder tab, you might need to refresh or restart the administration console.

• Unless you have reasons otherwise, use the example provided, and make sure you include the collection name.



Now that you've added reporting to the server and to the team project collection, you're ready to start adding reports to your projects.

# **Try this next**

Upload reports to a team project

# **Dig deeper**

- Do you want to know more about installing and configuring SQL Server for use with TFS? Read about <u>supported versions and editions</u>, <u>working with</u> <u>named instances</u>, <u>using SQL Server 2012 AlwaysOn</u>, as well as <u>understanding how TFS, SQL Server, and Reporting all work together</u>.
- Do you want to add someone as an administrator of a team project collection or for all of TFS? Learn what you have to do in <u>Set</u> <u>Administrator Permissions for Team Project Collections</u> and <u>Set</u> <u>Administrator Permissions for Team Foundation Server</u>.
- Do you need to change the data reader account you chose? Learn how to in <u>Change the Service Account or Password for SQL Server Reporting</u> <u>Services</u>, or read about <u>Service Accounts and Dependencies in Team</u> <u>Foundation Server</u> to better understand its requirements and what it does.



Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Upload reports to a team project
Send Feedback

By adding a report server to your TFS (on-premises) deployment, you can access a wealth of data about your team's projects, such as build quality, bug trends, burndown, and test progress. SQL Server Reporting Services (SSRS) reports provide insight to help teams manage work and improve processes.



After you've added a report server, you'll want to add reports to your team project. You can upload reports when connecting to an on-premises deployment of TFS 2010 or TFS 2012.

If the Reports page in Team Explorer appears empty, then you'll want to upload reports.



If the Reports page doesn't appear at all, then you'll want to add a report server.

## **Upload reports to your team project**

Use the **tfpt** command line tool that TFS Power Tools provides.

- 1. Verify that you have the following tools, configurations, and permissions.
  - If you haven't installed TFS power tools,

#### install them now.

• If you haven't installed Visual Studio or Team Explorer, <u>install one of them</u> <u>now</u>. You can download Team Explorer for free <u>here</u>. Team Explorer must be present on the same machine where you run the **tfpt** command line tool.

- If you aren't an administrator for the team project, get added now.
- If you haven't been added to the Content Manager or Team Foundation Content Manager role for SQL Server Reporting Services, get added now.

• If you haven't installed the latest <u>Quarterly Update for Microsoft Visual Studio</u> <u>Team Foundation Server 2012</u>, install it now. This is the only way to obtain the latest version of the default process templates, which contains the latest version of the reports.

• From Team Explorer, download the latest process template that is compatible with the one used to create your team project.



To determine the type of process template that was used to create your team project, review the work item types that appear in the New Work Item menu for Team Explorer and then compare them with the work item types in the following chart. If your work item types differ from those shown listed, then a custom process template might have been used.



#### ⊠Note

- In general, you can use the most recent version of the process template that was used to create your team project. For example, you can upload reports from Scrum 2.2 version, even if your team project was created with Scrum 2.0.
- To download a process template, you need to be an administrator for the team project collection, and you must connect to TFS using the same version of Visual Studio or Team Explorer. For example, connect to TFS 2012 from Team Explorer 2012.
- To learn more about process templates and work item types, go here.

• Open a Command Prompt in administrative mode and change to the directory where you installed the power tools.

#### Copy Code

cd %programfiles%\Microsoft Team Foundation Server 2012 Power Tools

On a 64-bit edition of Windows, replace %programfiles% with %programfiles(x86)%.

• Upload the reports. The process template you specify must be compatible with the one used to create the team project. And, it must have been uploaded to the team project collection that hosts your team project.

tfpt addprojectreports /collection:"http://MyServer:8080/tfs/Default

These are the names of the process templates uploaded with the latest quarterly update for TFS 2012 (later versions may be available):

- Microsoft Visual Studio Scrum 2.2
- MSF for Agile Software Development 6.2
- MSF for CMMI Process Improvement 6.2

To overwrite reports that were previously uploaded, specify the /force option.

• Open the report site from the 🗟 Reports page in Team Explorer. You might need to 🕸 refresh the page to see the newly uploaded reports.



And, here's a view of uploaded reports for a Scrum project in Report Manager.



Depending upon the amount of data that has been collected for your team

project, it can take several minutes to several hours for the data warehouse and cube to build. Once they do, however, you can view progress for your team project since TFS was first deployed.

## **Q** and A

## **Q:** How can I determine if my report is up to date?

A: To see when a report was last updated, check the Date Last Updated time stamp, which appears in the lower-right corner of each report.

### **Q:** How often are reports refreshed?

A: Reports access data from the relational data warehouse (Tfs\_Warehouse), which is updated every two minutes, and the cube (Tfs\_Analysis), which is updated every two hours.

#### **Q:** How do I resolve issues causing reports to be out of date?

A: Reports will be out of date when all or some data stops flowing into the data warehouse. To verify the data flow, run the GetProcessStatus of the <u>Warehouse</u> <u>Control Web service</u>. Also, fix any schema conflicts that may be occurring.

### **Q:** What tasks does tfpt addprojectreports perform?

A: The **tfpt addprojectreports** command performs the instructions contained in the ReportTasks file of the process template. This file is located in the Reports folder of the process template. The tasks include creating report folders, setting report parameters and data sources for each report, and uploading reports to the report server.

### **Q:** What if I just want to upload a single report?

A: You can upload a single report using Upload file in Report Manager. You'll need to define the report's data source and possibly other parameters.

#### **Q:** What are the names of the data sources?

A: The names of the reporting services data sources are Tfs2010ReportsDS and

Tfs2010OlapReportsDS. Even though you may have installed or upgraded to TFS 2012, these names, which were assigned to the data sources for a TFS 2010 installation, are in use.

You can always check the names by opening the context menu for any report, choose Manage, and then choose Data Sources.

### **Q:** Can I upload reports from another process template?

A: Most reports are customized based on the work item types and workflow states used in each process template. You might need to customize the reports you copy from another process template to fit the type definitions for your team project.

To learn about which reports are provided with which process templates, see <u>Review team activities to support useful reports</u>.

## **Q: What about uploading Excel reports?**

A: You can use **tfpt addprojectportal** command to add a project portal and Excel reports once you have added a SharePoint web application to your team project collection.

### **Q:** Can I get these same reports using Visual Studio Online?

A: Not at this time. Uploading reports is only supported for TFS on-premises deployments.

For additional questions, go to the Microsoft forum: <u>Team Foundation Server -</u> <u>Reporting & Warehouse</u>.

## **Try this next**

Grant permissions to view or create reports.

## **Related content**

- Do you need to customize a report? Download <u>SQL Server Report Builder</u> 2012.
- Do you need to learn more about how to create or customize TFS SSRS reports?
  - <u>Creating Reports for Team Foundation Server 2010</u>.
  - <u>Customizing Reports for Team Foundation Server 2010</u>.
- Do you want to learn more about the TFS cube? Perspectives and Measure Groups Provided in the Analysis Services Cube for Team System.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Grant permissions to view or create reports in TFS
Send Feedback

By adding a report server to your TFS (on-premises) deployment, you can access a wealth of data about your team's projects, such as build quality, bug trends, burndown, and test progress. SQL Server Reporting Services (SSRS) reports provide insight to help teams manage work and improve processes.



Now that you've uploaded reports, you'll want to enable members of your team to view or manage them. Also, to create or modify reports, you'll need to grant them access to read databases.

# ■Add accounts to predefined roles to view or manage reports

Add report viewers to the Browser role. Add TFS report authors to the Team Foundation Content Manager role.

#### ĭTip

Permissions to access Report Manager are managed separately from TFS permissions. Even if you have added team members to a TFS group, you will still have to add them to a Report Manager role.

- 1. If you haven't been added to the Content Manager role for Reporting Services, get added by someone who has been added to this role.
- 2. From the Report Manager home page, open Folder Settings.



The URL is http://ReportServer/Reports/Pages/Folder.aspx, or if using a named instance,

http://ReportServer/Reports\_InstanceName/Pages/Folder.aspx.

3. Open New Role Assignment.

SQL Server Home	Reporting Services	Home
	Delete     Mew Role Assignment     Group or User	Role(s)
Security	Edit       BUILTIN\Administrators         Edit       NORTHAMERICA\ctsodev1         Edit       NORTHAMERICA\ctsodev2         Edit       NT AUTHORITY\NETWORK         SERVICE       SERVICE	Content Manager Content Manager Content Manager Team Foundation Content Manager
ſ∎Tip		

To limit access to reports defined for a team project or team project collection, first navigate to the corresponding folder and then open New Role Assignment.

4. Add the account name and select their role.



# Add report authors to database roles

If members need to create or customize reports, add their accounts to the TfsWarehouseDataReader role. Report authors need read access to both the relational data warehouse and Analysis Services cube. Team members who create Excel reports from work item queries or by connecting to the cube need only read access to the cube.

- 1. If you aren't an administrator for the TFS database, get added as one.
- 2. Connect to the Database Engine for TFS using SQL Server Management Studio.



3. Open the properties page for the TfsWarehouseDataReader role under the

Databases/Tfs\_Warehouse/Security/Roles/Database Roles folder.

Object Explorer	
Connect 🗸 🛃 🛃 🔲 🍸 😰 😼	
<ul> <li>FABRIKAMPRIME (SQL Server 11.0.2218 - NORTHAN</li> <li>Databases</li> </ul>	MERICA\ctsoita)
<ul> <li>□ □ Tfs_Warehouse</li> <li></li></ul>	New Database Role Script Database Role as
□ □ Security	Policies  Facets
<ul> <li>Roles</li> <li>Database Roles</li> </ul>	Start Powershell Reports
accessadmin 🦓	Rename
ataRead	Delete Refresh
	Properties

4. Add the account.

	Dat	tabase Role Prop	erties - TfsW	/arehouseData	Reader				
Select a page General Securables Extended Properties		🔄 Script 👻 📑 Help							
		Role name: Owner:	TfsWareho dbo	useDataReader	r				
	2	Sele	ect Database	e User or Role	×				
Connection	Select the	ese object types:							
Server: FABRIKAMPRIME		atabase roles			Object Types				
Connection: NORTHAMERICA	Enter the	object names to se	lect ( <u>examples</u>	):					
		AMERICA\ctsopm		Check Names					
Progress					Browse				
Ready			ОК	Cancel	Help				
		K			Add Remove				

5. Next, connect to the Analysis Services database.

,P	Connect to Server	×
SOL Se	erver <sup>-</sup> 2012	
~ - <b>4</b>		
Server type	Analysis Services	~
Server name:	FABRIKAMPRIME	~
Authentication:	Windows Authentication	~
User name:	NORTHAMERICA\ctsopm	~
Password:		
	Remember password	

6. Open the properties page for the TfsWarehouseDataReader role under the Databases/Tfs\_Analysis/Roles folder.



7. Add the account.

<u>68</u>		Role Properties - TfsWarehouseDataReader							
Select a page General Membership Data Sources Cubes Cell Data Dimensions		Script   Help  The users and groups member of this role will have priviledges in all objects associated with this role.							
Mining Structur	2	Select Database User or Role ×							
Connection Server: FABRIKAMPRIME Connection: NORTHAMERICA	Select th Users, D Enter the	ese object types: Database roles Object Types object names to select ( <u>examples</u> ):							
View conn	NORTH	NORTHAMERICA\ctsopm Check Names							
Progress Ready		OK Cancel Help							
		Add							

#### **▲Important**

Accounts that you add to the TfsWarehouseDataReader roles can view data for all team projects that are hosted in all team project collections in the TFS deployment. There is no way to limit access to a team project or collection.

## **Try this next**

Review team activities to support useful reports.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Review team activities to support useful reports
Send Feedback

By adding a report server to your TFS (on-premises) deployment, you can access a wealth of data about your team's projects, such as build quality, bug trends, burndown, and test progress. SQL Server Reporting Services (SSRS) reports provide insight to help teams manage work and improve processes.



Now that you've uploaded reports, how do you use them to track progress, gain insight, and improve processes?

First, make sure your team is performing the activities that create the data that these reports use. Your team is probably performing most of these activities already.

Here's a summary of the reports that TFS provides and the team activities that are associated with them. Over time, you can use these reports to see trends and identify which practices and processes require more attention to deliver desired results.

# **Monitor code quality**

Build reports track the quality of software under development. By defining tests to run automatically as part of each build definition and instrumenting tests to gather code coverage data, you can gain insight about the quality of the builds, tests, and code.

Build and test activities 1. Configure a **Build** reports build system. • Build 2. Define your Quality build process. Indicators (Agile and 3. Run tests in CMMI your build only) process. • Build 4. Publish test **Success** results to TFS Over Time data store. (pictured) 5. Rate completed • Build builds to Summary populate the **Build Quality** 

For a free downloadable guide to testing and monitoring builds, see Testing for Continuous Delivery with Visual Studio 2012.

Sample build success over time report

dimension.

No Bu	ild Build F	ailed Build Succ No Te	eeded, sts	T	ests F	ailed		Test Low	ts Pas Cove	sed, rage		Pass	ed	
Build Definition	Platform	Configuration	6/2	6/3	6/4	6/5	6/6	6/9	6/10	6/11	6/12	6/13	6/14	6/15
AllDebug	x86	Debug												
DebugAnyCPU	Any CPU	Debug												
Debugx86	x86	Debug												
ReleaseAnyCPU	Any CPU	Release												
Releasex86	x86	Release												

## **Monitor progress**

Project management reports provide insight into how much work the team is tackling within a sprint or release, and the rate of their progress. By linking work items and updating specific fields as work is performed, you can track the progress of individual stories and be able to more accurately estimate future activities.

Work item tracking activities 1. Create the backlog. • Create product backlog items and specify the Project management (Scrum) reports Effort (Scrum). Backlog Overview • Create user stories and (similar to specify the Stories **Story Points** Overview) (Agile). • Release • Create Burndown requirements • Sprint and specify Burndown the Size (CMMI). • Velocity 2. Work in sprints. **Project management** Assign backlog

items to sprints, (Agile and CMMI) create tasks and reports link them to • Burndown and parent backlog Burn Rate items, and assign to a team • Remaining Work member. Requirements 3. Update Overview **Remaining Work** (CMMI) for tasks. For Agile and CMMI • Requirements team projects, Progress(CMMI) update **Completed Work** • Status of All as well. Iterations (similar to **⊿**Tip Velocity) The only report Stories that references Overview Original (Agile) Estimate is (pictured) Status of All Iterations. **Stories Progress** (Agile) 4. Create test cases • Unplanned Work and bugs, link them to their parent backlog item, and update their State. 5. (Optional) Assign work items to areas to filter reports.

#### Sample stories overview report



# **Monitor test plans and bug tracking**

Test planning reports support monitoring the test progress and coverage of backlog items or user stories. Bug tracking reports illustrate the team's capacity to find and resolve bugs.

Test planning and bug tracking activities 1. Define test plans and test cases, and Test and bug update their reports State as work progresses. 2. Mark the results of each validation step in manual tests as either passed or failed. 3. Create bugs, specify the Priority and Severity, assign to a team member, and update the State. 4. (Optional) Assign test

- Test Case Readiness
- Test Plan Progress (pictured)
  - Bug Status (Agile and CMMI only)
  - Bug Trends (Agile and CMMI only)
  - Reactivations (Agile and CMMI only)

cases and bugs to areas and iterations to filter reports.



Sample test plan progress report

## **Q** and A

• Q: Do reports handle stories and substories or tasks and subtasks?

A: Yes, you can subdivide stories or backlog items as well as tasks, creating a nested hierarchy of both backlog items and tasks. You can nest items several levels deep. If you subdivide a task into subtasks, specify hours only for the subtasks. These hours are rolled up as summary values for the parent task and their parent backlog item. To correct reports you believe are in error, see Address inaccuracies published for summary values.

• Q: Which reports depend on linking work items?

A: The overview and progress reports depend on linking tasks, test cases, and bugs to backlog items. You must link these items using the parent-child link for tasks and bugs and the Tested By link for test cases.

• Q: Which reports depend on Microsoft Test Manager?

A: Test Case Readiness and Test Plan Progress reports are designed to work with Test Manager. Also, the test points and test progress in project management overview reports depend on linking test cases to backlog items.

• Q: Are these reports the same as the agile planning charts that appear in Team Web Access (TWA)?

A: While some reports do display similar information, such as sprint burndown and velocity or status on all iterations, these reports are formatted differently and support additional filters. Other reports, such as the build and test planning reports, are not available through TWA at this time.

## **Related content**

- Create areas or iterations.
- Bulk edit work items to assign them to an area, iteration, team member, or priority.
- Add or modify a work item field to support reporting.
- <u>Creating Reports for Team Foundation Server 2010</u>.
- <u>Customizing Reports for Team Foundation Server 2010</u>.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic 
C#
Visual C++
F#
HLSL
JScript
Understanding Administrative Concepts and Tools
See Also Send Feedback

You can better manage Visual Studio Team Foundation Server (TFS) if you understand its architecture, the basic concepts behind its structure, and the tools for managing your deployment that are provided as part of the system. To deploy and manage TFS, you must understand both how it works and how it communicates with other components of Team Foundation. As an administrator for TFS, you must become familiar with Windows security restrictions, user account control, and SQL Server databases and security. If your deployment is integrated with SharePoint Products, you will also need to familiarize yourself with site collection and Web application management.

By choosing a deployment topology that best meets the needs of your business, you can help maintain the long-term manageability and sustainability of the software projects that you support. You can review the example topologies of TFS to see what kind of topology might best suit your needs.

TFS includes an administration console to help you with many of the administration tasks that are involved in managing and maintaining your deployment of TFS. By using the console, you can organize and manage users, collections of software projects, and build and virtual lab environments. You can also review the status of your deployment and view logs of server activity.
### **Common Tasks**

**Common Tasks** 

Supporting Content

Team Foundation Server Concepts

<u>Team</u> Foundation <u>Server</u> Architecture

<u>The Team</u> <u>Foundation</u> <u>Administration</u> <u>Console</u>

Learn about Team Foundation Server concepts and architecture: You can better plan for and manage TFS if you understand its architecture, the requirements for deploying it in a domain or a workgroup, and the basic concepts and terminology used throughout this

Naming Restrictions in Team Foundation

Using Team Foundation Server Command-Line Tools

Organizing Your Server with Team Project documentation.

**Collections** 

Interactions Between SharePoint Products and Team Foundation Server

Understanding SQL Server and SQL Server Reporting Services

Examples of Simple **Topology** 

Choose a deployment **Examples of** topology: You can install and configure Team Foundation Server in many ways. You can also expand or **Complex Topology** reduce your deployment as your **Examples of** needs change. Review the example topologies Hosted to help you decide what Topology configuration might Get Up and best suit your **Running With** development needs. a Single-Server Installation [Tutorial]

**Moderate Topology Examples of**  Maintain the health of your deployment: You can back up the databases that contain the data for your deployment of Team Foundation Server.

Backing up and Restoring Your Deployment

### **See Also**

### Concepts

<u>Configuring Your Server Using the Team Foundation Administration Console</u> <u>Organizing Your Server with Team Project Collections</u> <u>Configuring Users, Groups, and Permissions</u>

#### **Other Resources**

<u>Configuring Resources to Support Team Projects</u> <u>Managing Permissions</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic 
C#
Visual C++
F#
HLSL
JScript
Team Foundation Server Concepts
See Also Send Feedback

To deploy and manage Visual Studio Team Foundation Server effectively, you must understand how it works and communicates with other components of Team Foundation. As an administrator for Team Foundation Server, you should be familiar with Windows authentication, network protocols and traffic, and the structure of the business network on which Team Foundation Server is installed. You should also have an understanding of groups and permissions in Team Foundation Server. You might also find an understanding of SQL Server, SQL Server Reporting Services, and SharePoint Products useful as you manage Team Foundation Server.

# **Understanding Components and Terms**

You will be better able to plan, deploy, and manage Team Foundation Server if you understand the following components and terms:

- application tier, data tier, client tier: The logical tiers that compose Team Foundation Server. These tiers might all be deployed on the same physical computer, or they might be installed across multiple computers. For more information, see Examples of Simple Topology, Examples of Moderate Topology, Examples of Complex Topology, and Team Foundation Server Architecture.
- team project collection: The primary organizational unit for all data in Team Foundation Server. Collections can include one or more team projects. For more information, see <u>Organizing Your Server with Team Project</u> <u>Collections</u>.
- team project: A central point for your team to share team activities that are required to develop a specific software technology or product. Team projects are organized in team project collections. For more information, see Planning and Tracking Projects.
- Team Foundation Administration Console: The centralized management tool for administrators in Team Foundation Server. You can manage users and permissions from the administration console, in addition to creating and managing team project collections, adding and managing SharePoint Web applications for use in the deployment, creating and managing virtual labs, and reviewing server status. For more information, see <u>The Team</u> <u>Foundation Administration Console</u>.
- Service accounts: The account or accounts that the Web services and applications in Team Foundation use. Team Foundation Server requires service accounts to perform operations across servers and Web services. These service accounts have specific requirements. For more information, see <u>Service Accounts and Dependencies in Team Foundation Server</u>.

- SharePoint Products: Software that provides a scalable, manageable platform for collaboration and the development of Web-based business applications. You can include one or more SharePoint Web applications as part of your deployment of Team Foundation Server. To include one of these applications, you must install and configure Team Foundation Server extensions for SharePoint Products, and you must configure permissions across the deployment. For more information, see <u>Add a SharePoint Web</u> <u>Application to Your Deployment</u> and SharePoint Products and Technologies and Team Foundation Server.
- SQL Server and SQL Server Reporting Services: Software that provides a database platform for large-scale online transaction processing (OLTP), data warehousing, and e-commerce applications. SQL Server is also a business intelligence platform for data integration, analysis, and reporting solutions. Team Foundation Server stores its data in SQL Server databases. You can also optionally include a server that is running SQL Server Reporting Services and that automatically generates reports for team projects. For more information, see <u>Team Foundation Server Architecture</u> and Team Foundation Server Databases.

## **Understanding Team Foundation** Server Security

To optimize the security of Team Foundation Server, you should understand the following concepts:

- <u>Topology</u>, which includes where and how servers that are running components of Team Foundation are deployed, the network traffic that passes between Team Foundation Server and clients of Team Foundation, and the services that must run on Team Foundation Server.
- <u>Authentication</u>, which includes the determination of the validity of users, groups, and services in Team Foundation Server.
- <u>Authorization</u>, which includes the determination of whether valid users, groups, and services in Team Foundation Server have the appropriate permissions to perform specific actions.

You should also consider the other components and services on which Team Foundation Server depends.

When you consider security for Team Foundation Server, you must understand the difference between authentication and authorization. Authentication is the verification of the credentials of a connection attempt from a client, server, or process. Authorization is the verification that the identity that is trying to connect has permissions to access the object or method. Authorization occurs only after successful authentication. If a connection is not authenticated, it fails before any authorization checking is performed. If authentication of a connection succeeds, a specific action might still be disallowed because the user or group was not authorized to perform that action.

### **Topologies, Ports, and Services**

The first element of deployment and security for Team Foundation Server is whether the components of your deployment can connect to one another to communicate. Your goal is to enable connections between clients of Team Foundation and Team Foundation Server and to limit or prevent other connection attempts.

Team Foundation Server depends on certain ports and services so that it can function. You can secure and monitor these ports to help meet business security needs. You must permit network traffic for Team Foundation Server to pass between clients of Team Foundation, the servers that host the logical components of the application tier and the data tier for Team Foundation, computers for Team Foundation Build, and remote clients that are using Team Foundation Server Proxy. By default, Team Foundation Server is configured to use HTTP for its Web services. For a full list of ports and services that Team Foundation Server uses and how they are used within its architecture, see

#### Team Foundation Server Architecture.

You can deploy Team Foundation Server in an Active Directory domain or in a workgroup. Active Directory provides more built-in security features than workgroups provide. You can use Active Directory features to help secure your deployment of Team Foundation Server. For example, you can configure Active Directory to prevent duplicate computer names so that a malicious user cannot spoof the computer name with a rogue server that is running Team Foundation Server. To mitigate the same kind of threat in a workgroup, you must configure computer certificates.

Whether you deploy Team Foundation Server in a workgroup or a domain, you must comply with certain constraints imposed by the requirements of Team Foundation Server itself. For more information about topologies for Team Foundation Server, see Examples of Simple Topology, Examples of Moderate Topology, Examples of Complex Topology, Understanding Windows SharePoint Services, and Understanding SQL Server and SQL Server Reporting Services.

### Authentication

Security for Team Foundation Server is integrated with and relies on Windows integrated authentication and the security features of the Windows operating system. You can use Windows integrated authentication to authenticate accounts for connections between clients of Team Foundation and Team Foundation Server, for Web services on the servers that host the logical application and data tiers for Team Foundation, and for connections between application-tier and

data-tier servers themselves.

#### ⊠Note

You can configure Team Foundation Server to support Kerberos for mutual authentication of both the client and the server after you install Team Foundation Server.

You should not configure any SQL Server database connections between Team Foundation Server and SharePoint Products to use SQL Server Authentication because it is not as secure as Windows authentication. When you connect to the database, the user name and the password for the database administrator account are sent in an unencrypted format. Windows integrated authentication does not send the user name and password. Instead, it uses Windows integrated authentication security protocols to transfer service account identity information that is associated with the host Internet Information Services (IIS) application pool to SQL Server.

### Authorization

Team Foundation Server authorization is based on users and groups in Team Foundation, the permissions that are assigned directly to both those users and groups, and permissions that those users and groups might inherit by belonging to other groups in Team Foundation Server. Users and groups in Team Foundation can be local users or groups, Active Directory users and groups, or both.

Team Foundation Server is preconfigured with default groups at the server, collection, and project level. You can populate these groups by adding individual users. However, you might find management easier if you populate these groups by using Active Directory security groups. By taking this approach, you can manage group membership and permissions more efficiently across multiple computers or applications, such as SharePoint Products and SQL Server.

Your specific deployment might require that you configure users, groups, and permissions on multiple computers and within several applications. For example, you must configure permissions for users and groups in Reporting Services, SharePoint Products, and Team Foundation Server if you want to include reports and project portals as part of your deployment. In Team Foundation Server, you can set permissions for each project, for each collection, and across a deployment (at the server level). Additionally, certain permissions are granted by default to any user or group that you add to Team Foundation Server, as that user or group is automatically added to Team Foundation Valid Users. For more information about how to configure permissions, see

<u>Managing Permissions</u>. For more information about users and groups in Team Foundation Server, see <u>Configuring Users, Groups, and Permissions</u>.

Besides configuring permissions for authorization in Team Foundation Server, you might need authorization within version control and work items. You manage these permissions separately at a command prompt, but they are integrated as part of the interface for Team Explorer.

### See Also

### Concepts

<u>Team Foundation Server Architecture</u> <u>Configuring Users, Groups, and Permissions</u>

### **Other Resources**

### Managing Permissions Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Team Foundation Server Architecture
See Also Send Feedback

To best plan and manage your deployment, you should first understand the underlying architecture of Team Foundation Server (TFS). Understanding the architecture can help you maintain the overall health of the deployment and help ensure the overall availability of the servers and services your development teams require.

You can deploy TFS in several ways: on one server; on many servers; or in one domain or workgroup or across domains. Alternatively, you might choose to use Visual Studio Online, where all the server elements of your deployment are hosted for you by Microsoft. Understanding the architecture can help you decide what topology is most likely to meet your business needs. Regardless of your choice of topology, if you understand the architecture underlying TFS, you can better manage the physical and logical requirements. This topic provides a simple overview of the various architectures, with links to more information about example deployments. It also provides technical information about the services, databases, configuration information, and network ports and protocols of local deployments.

To understand the architecture of TFS and how it affects your deployment, you should consider the following:

- The logical application, data, and client tiers of Team Foundation, and whether you want to use one or more servers for the application and data tiers, or whether you want the application and data tiers hosted in the cloud for you by using Visual Studio Online
- The location of the physical or virtual servers that host those tiers
- Team Foundation Build and the number and location of build computers

that will run in your environment, including how many you might need to support your development practices

• The potential need for Team Foundation ServerProxy

In addition, you must consider the interactions between these entities. For example, if you choose to use the hosted TFS service, you must ensure that your clients can access the service on port 443. If you choose to deploy TFS locally, you must know what Web services, databases, and object models TFS uses. Also, you must know which network ports and protocols TFS uses by default and which network ports you can customize. Finally, you must understand what permissions you must set in Team Foundation Server and the components and programs on which your deployment depends.

Besides its own services, Team Foundation Server depends on other services in order to function. For information about these services, see <u>Team Foundation</u> <u>Server Concepts</u> and Components of the Data Warehouse for Team Foundation. For more information about the requirements and dependencies for installation, see Installing Team Foundation Server.

In this topic

- Visual Studio Online
- <u>The Object Model</u>
- Web Services and Databases for Local Deployments
  - Collection-Level Services
  - <u>Server-Level Services</u>
  - <u>Data Tier</u>
  - <u>Client Tier</u>
- Configuration Information
- Groups and Permissions

- <u>Network Ports and Protocols</u>
  - <u>Default Network Settings</u>
  - <u>Customizable Network Settings</u>

### **Team Foundation Service**



Microsoft offers the option of using Team Foundation Service, where all of the server-side aspects of the deployment are hosted for you. Your source code, work items, build configurations, and team features are all hosted in the cloud. From an architectural point of view, this greatly simplifies your deployment, as the only aspects of the architecture you need to consider are the client components and their Internet access.

When using the service, you use a web browser to connect to the service using your Microsoft account. You can create team projects, add members to your team, and work as you would with a locally installed deployment, without the overhead of administering the servers. Your application tier, data tier, and build servers are hosted for you in the cloud, using the Microsoft Cloud platform and SQL Server Azure.

For more information, see

Examples of Hosted Topology.

## The object model

With either the hosted or the locally-deployed architecture, you can extend the features and functionality of Team Foundation by writing an application that is based on its server or client object model. In all deployment types, you can write applications that extend client capabilities. However, if you want to extend server capabilities, your application must run on the application-tier server. To extend the client capabilities, you must run the application on the same computer as Team Explorer.



For more information, see Extending Team Foundation.

# ■Web services and databases for local deployments

Team Foundation Server includes a set of Web services and databases that you install and configure separately on the server or servers that host the logical application, data, and client tiers for Team Foundation. Some features, such as the task board, and backlog team-based features, are entirely web-based and accessed solely through Team Web Access, a client-side web based service. Others, such as the version control features, can be accessed through either Team Web Access or through a client application. The following illustrations provide a high-level view of web services, applications, and databases for local deployments of TFS.







### **Collection-level services**

Collection-level services provide the functionality for operations at the level of the team project collection. You can create applications that extend Team Foundation Server by using some of these services. For more information about creating applications for TFS, see Extending Team Foundation.

#### ĭTip

Some services appear in more than one level. For example, the Registry service functions at the collection level and the server level, and appears in both lists.

• Team Foundation Framework Services

- Registry service
- Registration service (for compatibility with earlier versions of Team Foundation Server)
- Property Service
- Event Service
- Security service
- Location service
- Identity Management service
- Version Control Web service
- Work Item Tracking Web service
- Team Foundation Build Web service
- Lab Management Web service
- VMM Administration Web service
- Test Agent Controller Web service

### **Server-level services**

Server-level services (also known as application-level services) provide the functionality for operations for Team Foundation Server as a software application. You can create applications that extend Team Foundation Server by using some of these services. For more information, see Extending Team Foundation.

- Team Foundation Framework Services
  - Registry service
  - Event service

- Team Project Collection service
- Property service
- Security service
- Location service
- Identity Management service
- Administration Service
- Collection Management Service
- Catalog Service

### ∃Data Tier

The data tier includes data, stored procedures, and other associated logic. When you use Visual Studio Online, the data tier is hosted for you using SQL Server Azure. In a local deployment of TFS, the logical data tier consists of the following operational stores within SQL Server. These stores might be located on one physical server or distributed across many servers. You can create applications that extend Team Foundation Server by using some of these operational stores. For more information, see Extending Team Foundation.

- Configuration database (TFS\_Configuration)
- Application warehouse (TFS\_Warehouse)
- Analysis Services database (TFS\_Analysis)
- Databases for team project collections (TFS\_CollectionName)

The following table provides a list of the databases that Team Foundation Server uses in local deployments. Unless otherwise noted, you can move all databases in this list from the original server and instance where they are installed and restore them to another server or instance.

Database Name and Description	Server
TFS_Configuration This database stores the catalog of resources and the configuration information for Team Foundation Server. This database contains the operational stores for Team Foundation Server.	Instance of SQL Server that is used when Team Foundation Server is installed and configured.
TFS_Warehouse This database stores the data for reports.	Instance of SQL Server that is used when Team Foundation Server is installed and configured.
TFS_Analysis This multi- dimensional database stores the aggregated data from team project collections.	Instance of SQL Server that is used when Team Foundation Server is installed and configured.
Databases for team project	Instance of SQL

collections One databaseServer thatfor each team projectiscollection, containing datacompatiblefrom all team projects inwith Teamthat collection.FoundationServer.Server.

### **Client Tier**

The client tier communicates with the application tier through the server object model, and uses the same Web services that are listed for that tier. This is true whether you deploy TFS locally, or if you use Visual Studio Online. Besides that model, the client tier consists of Visual Studio Industry Partners (VSIP) components, Microsoft Office integration, command-line interfaces, and a framework for check-in policies.

# **Configuration Information**

The hosted service depends on the client services, deployed locally, and an Internet connection to the application and data tiers hosted in the cloud. A local deployment of Team Foundation Server depends on SQL Server, Internet Information Services (IIS), and the Windows operating system. Contingent on your chosen topology, Team Foundation Server might also depend on SQL Server Reporting Services or SharePoint Products. Therefore, configuration information for Team Foundation Server can be stored in any of the following locations:

- IIS data stores.
- Configuration files for Team Foundation Server.
- Data sources for Reporting Services (for example, TFSREPORTS data).
- Configuration database for Team Foundation Server. The Team Foundation Server registry is part of the configuration database.
- Windows Registry.

For examples of different local deployment topologies and where these resources are stored, see

Examples of Simple Topology, Examples of Moderate Topology, and Examples of Complex Topology. As you maintain a local deployment of Team Foundation Server, you must take these configuration sources into account. To change the configuration in any way, you might need to modify information that is stored in multiple locations. You might also need to change configuration information for the data and client tiers. Team Foundation Server includes an administration console and several command-line utilities to help you make these changes. For more information, see <u>The Team Foundation Administration Console</u> and <u>Technical Reference for Team Foundation</u>.

### ■Synchronization of group identities between Active Directory and Team Foundation Server

In local deployments where Team Foundation Server is running in an Active Directory domain, group and identity information is synchronized when any of the following events occur:

- The application-tier server for Team Foundation starts.
- An Active Directory group is added to a group in Team Foundation Server.
- The period of time that is specified in the scheduled job elapses. The default is one hour, and all groups in Team Foundation Server update every 24 hours.

Identity Management Services (IMS) synchronizes with Active Directory, and changed identities propagate from the server to the clients. By default, all groups update within 24 hours, but you can customize this to better suit the needs of your deployment. For more information, see Trusts and Forests Considerations for Team Foundation Server. For local deployments that do not use Active Directory, see Managing Team Foundation Server in a Workgroup.

### **Groups and Permissions**

In a local deployment, Team Foundation Server has its own set of default groups and permissions that you can set at the project, collection, or server level. You can create custom groups and customize permissions at group and individual levels. However, users or groups that you add to Team Foundation Server are not automatically added to two components on which local deployments of Team Foundation Server can depend: SharePoint Products and Reporting Services. If your deployment uses these programs, you must add users and groups to them and grant the appropriate permissions before those users or groups will function correctly across all operations in Team Foundation Server. For more information, see

<u>Add Users to Team Projects</u>, Collaborate (dig deeper), <u>Configuring Users</u>, <u>Groups, and Permissions</u>, <u>Managing Permissions</u>, and Change permission levels.

For hosted deployments, access is controlled through a combination of Microsoft accounts and team membership. For more information, log onto the service with your Microsoft account and <u>choose Learn</u>.

### **Network Ports and Protocols**

By default, a local deployment of TFS is configured to use specific network ports and protocols. The following illustration shows network traffic for Team Foundation Server in a simple deployment.



Similarly, the hosted service for TFS is configured to use specific network ports and protocols. The following illustration shows network traffic in a hosted deployment.



The following illustration shows network traffic in a more complex deployment that includes the components for Visual Studio Lab Management.





### **Default Network Settings**

By default, communication between the computers in a deployment of Team Foundation uses the protocols and ports shown in the following table. If an asterisk (\*) follows the port number, you can customize that port.

Tier and service	Protocol	Port
Application tier - Web Services	HTTP/HTTPS	8080/443*
Application tier - SharePoint Products	HTTP	17012* if SharePoint Products was installed with Team Foundation Server;

Administration		otherwise, randomly generated	
Application tier - SharePoint Products and Reporting Services	HTTP Windows Management Instrumentation (WMI) service (required during installation to specify and verify the URLs for reporting services)	80* Dynamic port	
Data tier	MS-SQL TCP	1433*	
Data tier (SQL Server Analysis Services)	MS-AS	default (2382 or 2383)* The default port varies depending on the version of SQL Server you installed and the type of instance. Use SQL Server Configuration Manager to determine the ports used by	

		your deployment.
Team Foundation Server Proxy - client to proxy	НТТР	8081*
Team Foundation Server Proxy - proxy to application tier	HTTP/HTTPS	8080/443*
Client tier - Reporting Services	НТТР	80*
Client tier - Web services	HTTP/HTTPS	8080/443*
Build controller to application tier	HTTP/HTTPS	8080/443
Build agent to application tier	HTTP/HTTPS	8080/443
Test controller to application tier	HTTP/HTTPS	8080/443*

Application tier to test controller	.NET remoting	6901*
Application tier to Domain Name System (DNS)	DNS Dynamic Update	53
Application tier - Virtual Machine Manager	НТТР	8100
Test controller to test agent	.NET remoting	6910*
Test agent to test controller	.NET remoting	6901*
Build controller to build agent	SOAP over HTTP	9191
Lab agent to lab agent in an isolated environment	TCP sockets	9050
Build agent to build controller	SOAP over HTTP	9191

Virtual Machine Manager Administrator Console - Virtual Machine Manager	HTTP	8100
Virtual Machine Manager- Virtual Machine Manager hosts	Windows Remote Management (WinRM) to perform actions Background Intelligent Transfer Service (BITS) to transfer data	80 to perform actions 443 to transfer data
Virtual Machine Manager- Virtual Machine Manager library server	WinRM to perform actions BITS to transfer data	80 to perform actions 443 to transfer data
Application tier - Virtual Machine	Distributed Component Object Model/Windows Management Interface	135 Dynamically assigned in the range

Manager hosts	(DCOM/WMI) communication to transfer data	49152 to 65535
Client tier - Virtual Machine Manager hosts	Host-based connection to the virtual machine. For more information about host- based connections, see How to: Connect to machines in a Lab Environment.	2179 to perform host- based connections.
Hosted services	HTTPS	443

### **Customizable Network Settings**

As the previous table shows, you can change communication between the application, data, and client tiers in local deployments by modifying Team Foundation Server to use custom ports. The following table describes example changes in ports from HTTP to HTTPS.

#### ⊠Note

To configure Team Foundation Server to use HTTPS and Secure Sockets Layer, you must not only enable ports for HTTPS network traffic but also perform many other tasks. For more information, see

Securing Team Foundation Server.

Service	Protocol	Port
Web Services with SSL	HTTPS	Configured by the administrator
SharePoint Central Administration	HTTPS	Configured by the administrator
SharePoint Products	HTTPS	443
Reporting Services	HTTPS	443
Client Web Services	HTTPS	Configured by the administrator
### See Also

#### Concepts

Team Foundation Server Concepts Team Foundation Server Permissions Components of the Data Warehouse for Team Foundation Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
The Team Foundation Administration Console
See Also Send Feedback

By using the administration console for Team Foundation, you can manage resources and perform most management tasks for your deployment of Visual Studio Team Foundation Server (TFS) from a central location. For example, you can find all log files that detail system events in TFS. You can also perform the following tasks:

- manage users and permissions at the server level and the collection level
- create and manage team project collections
- add and manage SharePoint Web applications for use in the deployment
- create and manage virtual labs for testing your software
- add and manage build environments
- review server status

## **Using the Administration Console**

You can use the administration console for Team Foundation to perform most of the administrative tasks that are required to manage TFS. However, the functionality of the administration console depends on the permissions of the user who opens it. The console can be opened by anyone who has a valid user account on the server or servers on which the console is installed, but that does not mean that they will be able to use it to perform functions for which they do not have the required permissions. If a user opens the console but does not have the permissions that are required to perform certain functions within that console, the operation will fail when the user tries to perform that function. For more information, see <u>Open the Team Foundation Administration Console</u>.

The administration console is installed by default on any server on which you install one or more of the following components:

- an application tier for TFS
- TFS extensions for SharePoint Products
- Team Foundation Build
- Visual Studio Lab Management

For more information, see **Open the Team Foundation Administration Console**.

#### **Managing Server Resources**

If you open the administration console on an application-tier server, you can manage most aspects of the server resources for Team Foundation. In the Application Tier node, you can view summary information for the installation of TFS and manage server-level groups and security settings. In the Data Tier node, you can view information about the configuration database for TFS. In the Reporting Services node, you can view information about any resources in SQL Server Reporting Services that were configured for the deployment. For more information, <u>Review Server Status and Settings</u>.

In the Team Project Collections node, you can create, configure, view, start, and stop team project collections, which you can use to organize team projects that have similar needs or functionality. For more information, see <u>Organizing Your</u> <u>Server with Team Project Collections</u>.

In the SharePoint Web Applications node, you can configure one or more SharePoint Web applications for use with team project collections. If you add one or more Web applications, members of your team can create a SharePoint site automatically when they create a team project. Before you can add a SharePoint Web application, you must install the extensions for SharePoint Products on a server in the Web farm that hosts the application that you want to add. You must also add the service account that the SharePoint Web application uses to TFS. For more information, see Add a SharePoint Web Application to Your Deployment and Modify or Remove Access Between a SharePoint Web Application and Team Foundation Server.

In the Lab Management node, you can configure one or more virtual machines to act as resources for developing and testing your code. For more information, see How to: Configure or Reconfigure Lab Management and Managing Lab Services.

### ■Managing SharePoint Web Applications Used with Team Foundation Server

Before you can add one or more SharePoint Web applications to your deployment of TFS, you must first install the Team Foundation Server Extensions for SharePoint Products. Then you must configure access between the SharePoint Web application and TFS. Depending on your permissions, you either must configure access manually, or you can configure access automatically when you add the SharePoint Web application to TFS. After you grant access, you can manage the extensions if you open the administration console on a server on which the Team Foundation Server Extensions for SharePoint Products are installed. For more information, see <u>Add a SharePoint</u> <u>Web Application to Your Deployment</u> and Extensions.

### **Reviewing Log Files**

In the administration console, you can review the log files for installation and for

system changes. You can also open the folder where these files are stored and manually copy or delete them. For more information, see <u>Review Server Status</u> and <u>Settings</u>.

### **Managing Builds**

You can configure and manage Team Foundation Build so that your team can automatically and consistently build and test your software in a distributed environment. You can review and manage build settings by opening the administration console on the computers where Team Foundation Build has been installed. For more information, see <u>Configuring and Managing Your Build</u> <u>System</u>.

# ■Opening Specific Nodes in the Administration Console

The administration console contains different nodes based on what components of Team Foundation are installed on that server. If a particular node does not appear, you should verify that you are logged on to a server on which that component is installed.

### To open the node for Team Foundation Server

1. In the tree pane, expand the name of the server, and then expand the Application Tier node.

The summary information for the installation of TFS appears. Under this node, you can view information about the server and configure group membership, security settings, and service accounts.

- 2. (Optional) Under the Application Tier node, click one of the following nodes to view or modify the settings for that node:
  - Data Tier

View information about the server that hosts the configuration database for Team Foundation Server. You cannot modify this information.

• Team Project Collections

View, create, modify, or delete team project collections that are on this server. For more information, see <u>Organizing Your Server with Team</u> <u>Project Collections</u>.

• SharePoint Web Applications

View, create, modify, or delete a mapping from a server that is running Team Foundation Server to a server that is running SharePoint Products. For more information, see <u>Add a SharePoint Web</u> <u>Application to Your Deployment</u>.

• Reporting

View information about the server that hosts the warehouse database and the database for SQL Server Analysis Services. For more information, see <u>Add a Report Server to a Team Project Collection</u>.

• Lab Management

View or configure information about Lab Management. For more information, see How to: Configure or Edit Lab Management.

#### To open the node for Extensions for SharePoint Products

• In the tree pane, expand the name of the server, and then expand Extensions for SharePoint Products.

In this node, you can view, create, modify, or delete a mapping from a server that is running SharePoint Products to a server that is running Team Foundation Server. For more information, see <u>Add a SharePoint Web</u> <u>Application to Your Deployment</u>.

#### To open the node for Team Foundation Build

• In the tree pane, expand the name of the server, and then expand Team Foundation Build.

In this node, you can view, create, modify, or disable build services for Team Foundation Build. For more information, see <u>Configuring and</u> <u>Managing Your Build System</u>.

#### To open the Logs node

• In the tree pane, expand the name of the server, and then expand Logs.

In this node, you can review the logs of configuration changes in Team Foundation Server. For more information about a specific change, doubleclick it in the results pane.

### To open the Additional Tools node

• In the tree pane, expand the name of the server, and then expand Additional Tools.

In this node, the topnode remains static and does not display any information about installed tools in your deployment. However, if you install any additional tools from the Additional Tools and Components section of the Team Foundation Server Configuration Center, each tool should appear as a sub-node beneath this node. To view information about each installed tool, choose its sub-node underneath Additional Tools.

### See Also

#### Tasks

**Open the Team Foundation Administration Console** 

#### Concepts

Organizing Your Server with Team Project Collections Configuring Users, Groups, and Permissions

#### **Other Resources**

Get Up and Running With a Single-Server Installation [Tutorial] Configuring Resources to Support Team Projects

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Naming Restrictions in Team Foundation
See Also Send Feedback

Many components in Visual Studio Team Foundation Server (TFS) have certain naming restrictions. These restrictions help guarantee a consistent user experience and provide compatibility with other programs. These restrictions might include length, special characters, uniqueness, or other attributes.

This topic contains the following subsections:

Common Considerations

**Restrictions for Team Foundation Server Group Account Names** 

**Restrictions for Computer Names** 

**Restrictions for Team Project Collection Names** 

**Restrictions for Team Project Names** 

**Restrictions for Team Names** 

**Restrictions for Work Items** 

**Restrictions for Work Item Customizations** 

**Restrictions for Process Templates** 

**Restrictions for Team Foundation Build** 

**Restrictions for Version Control** 

See Also

### **Common Considerations**

The length restrictions in this topic are measured by the number of Unicode characters permitted. For more information about Unicode, see "About Unicode and Character Sets" (

http://go.microsoft.com/fwlink/?LinkId=76837). Surrogate characters are composed of two Unicode characters and these will count as two characters against the length restriction.

As with other operating system files, ASCII control characters (ASCII 1-31) and surrogate combinations are also not allowed. For general information about the operating system restrictions applied to file names, see "<u>Naming Files, Paths</u>, <u>and Namespaces</u>" ().

### ■Restrictions for Team Foundation Server Group Account Names

Team Foundation group accounts enable you to apply certain rights or permissions to a group of users. This Team Foundation group can consist of Windows user accounts, Windows group accounts, Active Directory group accounts, Team Foundation group accounts, or any mixture of these types. For more information about managing users with groups, see Users, groups, and permissions in Team Foundation Server deployments and

<u>Configuring Users, Groups, and Permissions</u>. If you want examples of how to add a new member to groups used in TFS, SharePoint Products, and SQL Server Reporting Services, see <u>Add Users to Team Projects</u>. If you want to know more about how to check the permissions for your account, see Change permission levels.

When a Team Foundation group account is created or changed, it must meet certain Team Foundation Server restrictions. The following table describes these restrictions.

Restriction Type	Restriction
Group • account name length	Must not contain more than 256 Unicode characters
Uniqueness (collection- level group accounts)	Must not match any other name of a group account in the team project collection

Uniqueness (project-level group accounts)	<ul> <li>Must not match any other group name in the same project</li> </ul>
Reserved group names	<ul> <li>Must not be named \$NAMESPACE at either the project or the server level</li> </ul>
	<ul> <li>Must not include the following printable characters: "/ \ [ ] :   &lt; &gt; + = ; ? *</li> </ul>
Special character restrictions	<ul> <li>Must not include nonprintable characters in the ASCII value range of 1-31</li> </ul>
	<ul> <li>Must not end in a period (.)</li> </ul>
	• Must not include commas (,)
<b>⊠Note</b>	

You do not create user accounts in Team Foundation Server.

## **Restrictions for Computer Names**

During the TFS installation process, the computer name is associated with the name of the server.

Both the operating system and Active Directory impose certain restrictions on computer names. For more information about renaming a computer, see "Rename a Computer" (

http://go.microsoft.com/fwlink/?LinkId=76839). For more information about Active Directory, see "Windows Server 2003 Active Directory" (http://go.microsoft.com/fwlink/?LinkId=47541).

## ■Restrictions for Names of Team Project Collections

The name of a team project collection identifies a grouping of team projects and the resources that are associated with those projects. A team project collection is an organizing structure that you can use to define and control a group of team projects within TFS. Team members will use the name of the team project collection when they connect to team projects in TFS. For more information, see

Organizing Your Server with Team Project Collections.

The following table describes the restrictions for names of collections.

Restriction Type	Restriction
Length	<ul> <li>Must not contain more than 64 Unicode characters</li> </ul>
	<ul> <li>Must not be identical to any other collection name in your deployment of Team Foundation Server.</li> </ul>
Uniqueness	<ul> <li>If your deployment includes SharePoint Products or SQL Server Reporting Services, must not be identical to the name</li> </ul>

	and full path of an existing SharePoint site, report server, or Reporting Services Web site.
	• Must not contain any Unicode control characters or surrogate characters.
Special	<ul> <li>Must not contain the following printable characters: /:\~&amp;%; @'"?&lt;&gt; #\$*}</li> <li>{, + = []</li> </ul>
Characters	<ul> <li>Must not contain an ellipsis () or a double period ()</li> </ul>
	• Must not start with an underscore (_)
	• Must not start or end with a period (.)
	• Must not be a system- reserved name such as PRN, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, COM10, LPT1,
Reserved names	LPT5, LPT6, LPT7,

LPT8, LPT9, NUL, CON, AUX, Web, or WEB

For more information about reserved names, see this page on the Microsoft Web site: <u>File Names, Paths,</u> <u>and Namespaces</u>.

## **Restrictions for Project Names**

The project names in TFS identify a collection of work items, documents, reports, team builds, and a version control tree that make up a particular project in TFS. Team members will use this project name to connect to the project in TFS.

The following table describes the restrictions for project names.

Restriction Type	Restriction
Length	• Must not contain more than 64 Unicode characters
Uniqueness	Must not be identical to any other name in the team project collection, the SharePoint Web application that supports the collection, or the instance of SQL Server Reporting Services that supports the collection
	<ul> <li>Must not contain any Unicode control characters or surrogate characters</li> <li>Must not contain the following printable</li> </ul>

I

Special characters •	<pre>characters: / : \ ~ &amp; % ; @ ' " ? &lt;&gt;   # \$ * } { , + = [ ] Must not contain an ellipsis () or a double period ()</pre>
•	Must not start with an underscore (_)
•	Must not start or end with a period (.)
• Reserved names	Must not be a system- reserved name such as PRN, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM6, COM7, COM8, COM9, COM10, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, NUL, CON, or AUX For more information about reserved names, see "

Naming Files, Paths, and Namespaces".

### **Restrictions for Team Names**

The Team names in TFS identify a group of individuals or groups that collectively work together as a team in a project in TFS. Team members will use this name to connect to the team in Team Web Access. The team name must be a name that can be rendered as part of a valid URL. In addition, each team name must be unique within a single project, but there is no restriction barring using the same team name in different projects within a team project collection. For more information about working in teams in TFS, see Collaborate (dig deeper).

The following table describes the restrictions for team names.

Restriction Type	Restriction
Length	• Must not contain more than 64 Unicode characters
l Uniqueness a I	Must not be identical to any other name in the project
	• Must not contain any Unicode control characters or surrogate characters
Special characters	<ul> <li>Must not contain the following printable characters: /:\~&amp;%; @'"?&lt;&gt; #\$*}</li> <li>{, +=[]</li> </ul>

•	Must not contain an ellipsis () or a double period ()
•	Must not start with an underscore (_)
•	Must not start or end with a period (.)
Reserved names	Must not be a system- reserved name such as PRN, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM6, COM7, COM8, COM9, COM10, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, NUL, CON, or AUX

Naming Files, Paths, and Namespaces.

### **Restrictions for Work Items**

Microsoft Visual Studio Application Lifecycle Management tracks the progress on a project by using items such as bugs, requirements, tasks, and risks. These items are referred to generically as work items. This section describes restrictions on the data stored in the work items.

### **Restrictions for Work Item Attachments**

Files can be attached to work items. The following table describes the restrictions on work item attachments.

Restriction
• By default, the work item file attachment size limit is 4,096 kilobytes. This value can be changed by the Team Foundation Administrator. The maximum size that can be set by the Administrator is 2 gigabytes.
• Default size: 4,096 kilobytes
• Maximum size: 2 gigabytes

### **Restrictions for Work Item Areas and Iterations**

Work items contain a field for Project Area and a field for Project Iteration. They are used to organize and display work items into logical groupings.

The Project Area and Project Iteration are paths made up of multiple node items separated by backslash (\) characters. Nodes are defined by the Team Foundation Server administrator to reflect the project areas and project cycle. The following table describes the restrictions on nodes and paths.

Restriction
• Must not contain more than 255 characters
• Must not contain Unicode control characters
<ul> <li>Must not contain any one of the following characters: \ / \$ ? * : " &amp; &gt; &lt; # %  </li> </ul>
• Must not contain characters prohibited by the local file system. For more information about Windows character restrictions, see

Naming Files, Paths, and Namespaces.

Reserved names

- Must not consist only of a period (.) or two periods (..)
- Must not be a system-reserved name such as PRN, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, COM10, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, NUL, CON, or AUX
- For more information about reserved names, see <u>Naming Files</u>, <u>Paths</u>, <u>and</u> <u>Namespaces</u>.

Path length

• Must not contain more than 4000 Unicode characters

Path hierarchy depth

• Must not be more than 14 levels deep

# Restrictions for Work Item Customizations

Team Foundation Server tracks the progress on a project by using items such as bugs, requirements, tasks, and risks. These items are referred to generically as work items.

Administrators of team projects can decide to change work item type definitions either at the project level or in a process template. For more information about how to customize work item types, see Customize Project Tracking Data, Forms, Workflow, and Other Objects. This section describes restrictions you will encounter when you customize work items and their associated elements.

### **Restrictions for Work Item Field Names**

Each work item type contains one or more work item fields. These fields define the information stored in a work item type. A work item field name uniquely identifies each work item field.

The following table describes the restrictions for work item field names.

Restriction Type	Restriction
Length	Must not contain more than 128 Unicode characters
	<ul> <li>Must not contain leading or trailing spaces</li> </ul>
	• Must not contain two

Special	or more consecutive spaces
enaracters •	Must not contain periods (.)
•	Must not contain opening or closing square brackets ([ ])
• Scope	Work item field names are scoped to an entire deployment of TFS. If a field name is changed, the change will be reflected in all projects and the work item types that contain that work item field.

### **Restrictions for Work Item Field Reference Names**

Each work item field has an associated field reference name. The field reference name uniquely identifies each field and cannot be changed after it is assigned. The following table describes the restrictions applied to field reference names.

Restriction Type	Restriction
Length	<ul> <li>Must not contain more than 70 Unicode characters</li> </ul>

•	Must not contain hyphens (-)
•	Must contain at least one period (.)
Special • characters	Must not start or end with a period (.)
•	Must not start with a number
•	Must not start with an underscore (_)
•	Must not be identical to any other field reference name in Team Foundation Server
• Uniqueness	Must not be identical to any other field reference name after those names are processed by the computer to replace all periods (.) with underscores (_)
	For example, the field reference names MyField and MyField would both be processed by the computer to be the same name:

#### My\_\_\_Field

### ■Restrictions for Work Item Field Help Text

As an option, you can associate help text with work item fields by using the <HELPTEXT> tag. The system displays this text at run time to help users know what to enter into the field. For more information about work item field help text, see Define the Help Text for a Work Item Field.

The following table describes the restrictions for work item field help text.

Restriction Type	Restriction
Length	• Must not contain more than 255 Unicode characters
Scope	Unlike the field name and field type, field Help text is scoped to a specific work item type in a specific team project.

### **Restrictions for Global Lists**

A global list is a set of list item values that is stored and used globally by all Team Foundation servers in a Team Foundation Server implementation. As you define work item types, you may find that some work item fields share the same set of possible values. Global lists enable you to define these values one time and share them among multiple work item types. For more information, see Define Global Lists.

A global list (GLOBALLIST) contains one or more list items (LISTITEM

elements).

L

The following table describes the restrictions on list items.

Restriction Type	Restriction
Length	<ul> <li>Must not contain more than 254 Unicode characters</li> <li>Must not be empty</li> </ul>
Special characters	<ul> <li>Must not contain leading or trailing white space</li> <li>Must not contain two consecutive spaces</li> </ul>
	<ul> <li>Must not contain backslash (\) characters</li> </ul>
Scope	• Since global lists are available among all projects, they must not contain elements defined at the project level, such as project- specific group account definitions.

The following table describes restrictions that apply to a global list.

Restriction Type	Restriction
Number of items	The global list must not be empty. It must contain at least one LISTITEM element.
Uniqueness	• The global list name must not be identical to any other global list name in Team Foundation Server.

### **Restrictions for Process Templates**

A process template is a set of default work items, work item queries, product templates, reports, security groups, and guidance that influences the structure of a project in Team Foundation. Team Foundation Server includes two default process templates that encompass two different styles for managing the software cycle. These templates can be customized to reflect the unique needs of your organization. For more information, see Customize Process Templates.

The following table describes restrictions on the process templates.

Restriction Type	Restriction
Process template name length	Must not contain more than 256 Unicode characters.
Process template name uniqueness	<ul> <li>Must be unique across the Team Foundation server.</li> <li>If you upload a template with the same name as an existing template, the existing template will be overwritten.</li> </ul>
Process template file size	The process template file size must not exceed 2 GB (gigabytes).

|

## ■Restrictions for Team Foundation Build

Team Foundation Build lets you manage all the aspects of the build process on a single computer. By using Team Foundation Build, you can synchronize the sources, compile the application, run associated unit tests, perform code analysis, release builds on a file server, and publish build reports.

### **Build Computer Restrictions**

Team Foundation Build is a separate installation from the Team Foundation Server application tier, data tier, or Visual Studio client. You may designate a separate computer. Otherwise, you can install the build side-by-side on the client computer or on the servers.

The following table describes restrictions for the build computer.

Restriction Type	Restriction
Disk space	Must contain sufficient space for the build (insufficient space will lead to failed builds).
Build directory	Must be a local path (for example, C:\builddirectory).
Drop location directory	Must be a UNC path (for example, \\server\share).

Each generated build is put in a new directory in the drop folder. • The Team Foundation Server Service account Drop (for example, location Domain\TFSSERVICE) permissions must have Full Control permission access to the UNC drop location. • The UNC drop location must be a shared folder. If you change the TFS Service account after the initial installation, you must make sure that the following conditions are true. • The account is a member of the Build Team Services group. Foundation Build The account has • Service read/write permissions account to the temporary folders and the ASP.NET temporary folder. The account has Full • Control permission to the build directory and drop location. If the build computer is

Firewall	firewall enabled, make sure
issues	that the program
	tfsbuildservice is in the
	exceptions list.

### **Build Type Names**

Team Foundation Build uses build types to configure the conditions under which a single solution or a set of solutions in a team project will be built. To conduct a build, you must either create a new build type or use an existing build type. For more information about build types, see Define Your Build Process.

The following table describes the restrictions on build type names.

Restriction Type	Restriction
l Uniqueness a i	Must not be the same as any other build type name n the project
Special characters	<ul> <li>Must not contain the following printable character: \$</li> </ul>

### **Build Quality Names**

The build quality lets you attach a quality level or completion state to a completed build. Team Foundation Build also lets you create new values for the build quality type. For more information, see Create a Build Definition. For a list of the default build quality values, see Rate the Quality of a Completed Build.

The following table describes the restrictions on build quality names.

Restriction Restriction Type

Must not contain more Length than 256 Unicode characters

Must not be the same as any other Build Quality Uniqueness name in the Team Foundation Build computer
# **Restrictions for Version Control**

Team Foundation version control provides a central repository for files and the commands that are required to manage those files across a team. It also provides customizable check-in policies, branching, merging, shelving, and many other features.

### **Version Control Paths**

The following table describes restrictions on a version control path.

Restriction Type	Restriction
	• Must not contain more than 259 Unicode characters for a single folder or file name.
Server source control folder path length	<ul> <li>Must not contain more than 388 Unicode characters for a directory.</li> <li>Must not contain more than 399</li> </ul>
	for a combined directory and file name.
	• Must not contain more than 248

Local	Unicode characters for a directory.
folder path • length	Must not contain more than 260 Unicode characters for a combined directory and file name.

See also: Optimize Your Workspaces.

### **Adding Files into Version Control**

The version control system stores many different types of files. For more information about how to add existing Visual Studio projects or solutions into version control, see Set Up Your Dev Machine and Start Using Version Control.

The following table describes the restrictions applied to files and folders to be added to version control.

Restriction Type	Restriction
• File extensions	Must not contain the following printable characters: / \ : * ? " < >  ;
•	Must not be empty
• Folders	Must not contain files that have pending changes



### **Label Names**

In Team Foundation version control, a label is a name applied to a specific set of revisions. You can attach labels to a set of unrelated files in version control. This lets you retrieve the files or act upon them as a group. The following table describes the restrictions put on label names.

Restriction Type	Restriction	
Length	Must not contain more than 64 Unicode characters	
Special characters	<ul> <li>Cannot end with a space () or period (.)</li> <li>Must not contain the following printable characters: "/:&lt;&gt;\  *?@;</li> </ul>	

### **Shelvesets**

Shelvesets enable you to set aside temporarily a batch of pending changes and then, as an option, remove the pending changes from your workspace. Later, you can restore the changes in a shelveset to your workspace or put them into another user's workspace.

The following table describes restrictions on shelveset names.

Restriction Type	Restriction
Length	Must not contain more than 64 Unicode characters
Special characters	<ul> <li>Must not contain the following printable characters: " / : &lt; &gt; \   * ?;</li> </ul>

### **Workspace** Names

A workspace is a client-side copy of the files and folders in Team Foundation version control. When you create multiple workspaces, you can have different versions of the same version control folder on a client computer. For more information about workspaces, see Manage Your Workspaces. The following table describes restrictions on workspace names.

Restriction Type	Restriction
Length	Must not contain more than 64 Unicode characters

- Cannot end with a space ()
  Special characters
  Must not contain the following printable
  - following printable characters: " / : < > \ | \* ? ;

## **See Also**

#### Tasks

Create a Build Definition Rate the Quality of a Completed Build

#### Concepts

Customize Project Tracking Data, Forms, Workflow, and Other Objects Define the Help Text for a Work Item Field Define Global Lists Customize Process Templates Define Your Build Process Manage Your Workspaces Send feedback on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Examples of Simple Topology
See Also Send Feedback

You can install and configure Visual Studio Team Foundation Server in several topology configurations. Generally speaking, the simpler the topology, the more easily you will be able to maintain a deployment of Team Foundation Server. You should deploy the simplest topology that meets your business needs. This topic describes two fairly simple topologies, in which the server and clients are all contained within a single workgroup or domain.

# **Simplest Topology**

The simplest server topology will use the fewest number of physical servers to host the components that compose the logical tiers of Team Foundation. The following illustration shows the simplest topology:



In this example, all server components are deployed on a single physical server. You can access them from client computers in the same domain or workgroup. This example is designed for a small product development team that has fewer than 50 users.

In this configuration, you can install the computer that is running Team Foundation Build and the team's test components on either the single server, which is running Team Foundation Server, or on one or more client computers. This configuration is best suited to small development organizations or pilot projects within larger organizations.

# **Simple Topology**

The simple server topology will also use the fewest number of physical servers to host the components that compose the logical tiers of Team Foundation. However, this topology also recognizes the additional load that building and testing software places on processing power. The following illustration shows a simple topology for Team Foundation Server:



In this example, the Web services and databases for Team Foundation are hosted on the same physical server, but the build services are installed on a separate computer. You can access Team Foundation Server from client computers in the same domain or workgroup. This example is designed for a small product development team that has fewer than 100 users.

In this configuration, you install the computer that is running Team Foundation Build and the team's test components on a computer that is dedicated to that purpose. This configuration is best suited to smaller development projects where builds and testing demands are regular and performance is a greater concern.

## See Also

### Concepts

Examples of Moderate Topology Examples of Complex Topology Team Foundation Server Architecture

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Examples of Moderate Topology
See Also Send Feedback

You can configure Visual Studio Team Foundation Server in several topology configurations. Generally speaking, the simpler the topology, the more easily you can maintain a deployment of Team Foundation Server. You should deploy the simplest topology that meets your business needs. This topic describes a moderately complex topology, in which the logical components of the data and application tiers of Team Foundation are installed on separate physical servers. Client computers within the trusted domains can access Team Foundation Server.

# **A Moderate Server Topology**

A moderate topology will use two or more servers to host the logical components of the data and application tiers of Team Foundation. The following illustration demonstrates a moderately complex topology for Team Foundation Server:



In this example, the services for Team Foundation Server are deployed on one server, referred to as an application-tier server, and the databases for Team Foundation Server are installed on a separate server, referred to as a data-tier server. A separate server hosts the SharePoint Web application that Team Foundation Server uses, and another server hosts the instance of SQL Server Reporting Services that Team Foundation Server uses. The portal for each team project is hosted on the SharePoint Web application. Therefore, the administrator must configure permissions for the users of that project on that Web application. The same consideration applies to configuring permission for users in SQL Server Reporting Services. Team Foundation Build and the team's test controllers are deployed on additional servers. In this illustration, the domain for the Cleveland clients is a child domain of the parent domain in Seattle. The child domain has a two-way transitive trust relationship with its parent domain. The service account for Team Foundation Server is trusted by both domains. Users in the child domain can access the server, and they are authenticated automatically by Windows integrated authentication. In this configuration, Team Foundation Server Proxy is required and installed in the Cleveland office. This example is designed for a moderately sized product development team that has fewer than 1,000 users.

## See Also

### Concepts

Examples of Simple Topology Examples of Complex Topology Team Foundation Server Architecture

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Examples of Complex Topology
See Also Send Feedback

You can configure Visual Studio Team Foundation Server in one of several topology configurations. Generally speaking, the simpler the topology, the more easily you will be able to maintain a Team Foundation Server deployment. You should deploy the simplest topology that meets your business needs.

This topic describes two complex Team Foundation Server topologies. A complex topology has Team Foundation Server components distributed across many servers. It is accessed by clients in multiple domains in geographically distant physical locations. Team Foundation Server Proxy is installed in addition to optional components such as SharePoint Products and SQL Server Reporting Services. There are several different code bases, and these code bases each have their own team project collection for the projects that use them. The databases for team project collections are stored on several different servers that are running SQL Server.

# **Complex Single-Domain Topology**

A complex server topology uses multiple servers in clusters or other failover configurations to host the logical Team Foundation data and application tier components. The following diagram demonstrates a complex single-domain topology:



This example topology is similar to the moderate topology (see Examples of Moderate Topology). Team Foundation Server services are deployed on one server and the Team Foundation Server databases are installed on a separate server, with Team Foundation Build and the team's test agents and test agent controllers deployed on additional servers. Also, fail-over components have been added. The Team Foundation databases have been installed on a SQL Server

cluster.

The example diagram shows child domains in Seattle and Cleveland, each with a two-way transitive trust to the parent domain. The service account for Team Foundation Server is trusted by both domains. Each geographically distant child domain uses a limited bandwidth connection. Since getting an enlistment from the version code control component of Team Foundation Server can take a very long time under such limited bandwidth conditions, a Team Foundation Server Proxy is installed in each child domain to act as a version control file proxy cache. All client requests go directly to the application tier, with the exception of version control code **get** requests. These requests are directed through the Team Foundation Server Proxy, which then caches all source files that it downloads on the proxy server.

The topology in this example is also concerned about fault tolerance and high performance. Data tier fault tolerance is provided by taking advantage of the SQL Server clustering technologies. Multiple SharePoint Web applications are configured for use by team projects within collections in a SharePoint Web farm. Team project collection databases are distributed across SQL Server instances for improved performance and ease of individual management. SQL Server Reporting Services and SQL Server Analysis Servers are running on separate servers for improved performance.

This example is designed for a large product development team with up to 2,000 users.

# **Complex Multi-Domain Topology**

A complex multi-domain server topology uses multiple servers in two or more domains. As with the complex single-domain topology, the deployment uses clusters or other failover configurations to host the components of the data tier for Team Foundation. The administrators for this deployment have configured network load balancing and added multiple application-tier servers to distribute the operational load. The following illustration demonstrates a complex multidomain topology:





As in the above example, this topology is configured for fault tolerance and high performance. In addition, this topology is distributed across multiple domains, some of which are fully trusted child domains of the parent domains, but one of which (IsolatedLab.com) is a completely separate domain. The service accounts used by Team Foundation Server are fully trusted members of all domains, and user accounts have been configured in both the parent domain and the separate domain as necessary for users who must work in both domains. Firewalls have been configured to allow traffic across the ports required by Team Foundation Server.

## See Also

### Concepts

Examples of Simple Topology Examples of Complex Topology

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Examples of Hosted Topology
See Also Send Feedback

Instead of installing and configuring Visual Studio Team Foundation Server (TFS) locally, Microsoft offers you the option of using Team Foundation Service Preview, where Microsoft hosts all the server elements of the deployment for you in the cloud. This greatly simplifies your deployment, because the only aspects of the architecture you need to consider are making sure your users have the Internet access and the permissions that are required to access your project.

# **The hosted environment**



Non-Microsoft trademarks displayed in this image remain the property of their respective owners.

If you choose to use the service, your entire project, including source code, work items, build configurations, and team features are all hosted in the cloud. From C# to Python, from Windows to Android, you can use a variety of languages and target a variety of platforms, and you can use a variety of tools. The only aspects of the architecture you need to manage are the access requirements.

When you use Team Foundation Service Preview, you use a web browser and your Microsoft account to connect to the hosted service. You can create team projects, add members to your team, and work just like you would work with a locally installed deployment, but without the overhead of administering the servers. Your application tier, data tier, and build servers are hosted for you using the Microsoft Cloud platform and SQL Server Azure. This is a great architecture to choose if you want the features of TFS, such as source control, work items, and agile project management, but do not want to support managing the physical infrastructure. It is also a great option to consider if your team members are distributed around the world. Because the service is entirely web-based, the only aspects you must manage are whether your team members have access to the Internet, whether their firewalls block port 443, and whether a Microsoft account can be added for each team member to each of the projects that are hosted for you.



Because the server resources for Team Foundation Service Preview are managed for you, it can be a good choice for teams that have unpredictable usage spikes or that might need to greatly expand or contract the resources for a project. The service infrastructure is designed to absorb fluctuating needs, and can seamlessly expand to manage the largest projects. It is also a great choice for teams with little or no IT support resources.

Unlike local deployments, the hosted service does not support integration with SharePoint Products or SQL Server Reporting Services, and does not fully integrate with Visual Studio Lab Management. Because of this limitation, the hosted service is not a good choice if your team needs detailed reports on progress-which is available in the integration between TFS and SQL Server Reporting Services-or requires virtual lab machines and automated testing, which is offered by Visual Studio Lab Management. Although the service provides a team page and dashboard, it is not as detailed as the integration available between TFS, SQL Server Reporting Services, and SharePoint Foundation 2010.

In addition, Team Foundation Service Preview is not a good choice if your project has strict governance requirements. Do not choose the hosted service if you are developing in a strictly regulated environment, such as the financial industry, or if your project must have complete control of its infrastructure for auditing purposes.

### **Examples of hosted deployments**

You can choose to have your build machine resources hosted in the cloud along with the rest of the server-side aspects of your deployment, or you can choose to use local build servers. You might choose to use a hosted build server for integrated builds, while your developers use local build servers to run interim





For more information about the service and options for using it, see <u>Team</u> <u>Foundation Service Preview</u>.

## See Also

### Concepts

Team Foundation Server Architecture Installing Team Foundation Server Planning and Tracking Projects

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Managing the Server Configuration
See Also Send Feedback

Depending on how you initially installed and configured Visual Studio Team Foundation Server (TFS), you might need to manage it across one or multiple servers to maintain and operate your deployment. For example, the following illustration shows the logical architecture of a deployment of TFS that is integrated with both SQL Server Reporting Services and SharePoint Products:



As the previous illustration shows, you would need to manage the configuration of the components, the users, the groups, and permissions across several servers. You must ensure that none of the servers had firewalls or other devices that blocked network traffic on the ports that are required for communication between the servers. You must ensure that all services that the deployment requires are running and configured to start automatically. Lastly, you must ensure that the service account that TFS uses is fully trusted by every computer in the deployment.

You can add or remove resources to your deployment to better meet the changing needs of your business and the software projects that you support with TFS. You can add or remove reporting and Web resources at any time. You can also use more than one instance of SQL Server to host the databases for your

deployment. For example, you can add a server that is running SQL Server Reporting Services to your deployment after you install and initially configure TFS. You can also upgrade the version of SharePoint Products that supports your deployment and add its capabilities of that product to the team projects that already exist in your deployment.

The administration console for Team Foundation can help you manage and monitor your server configuration, and you can also use the command-line tools that TFS includes. Both the console and the tools are installed as part of TFS.

## -Common Tasks

**Common Tasks** 

**Supporting** Content

Manage the servers and resources in the deployment: After you install and initially configure TFS, you can monitor and change the configuration to support your development projects.

• Add and manage SharePoint Web applications: You can support your team projects with resources in SharePoint Products by installing the Team Foundation Server extensions for SharePoint Products on one or Modify or more servers and adding SharePoint Access Web applications as resources that are available for **Application** team projects and and Team

**Configuring** Your Server Using the Team Foundation Administration Console Add a **SharePoint** Web Application to Your **Deployment** <u>Remove</u> Between a **SharePoint** Web

team project <u>Foundation</u> collections. <u>Server</u>

- Add and manage a <u>Change access</u> reporting resource: levels You can support your team projects Add reports to a team project with reports and reporting abilities by adding a report **Configuring** Lab server. After you add a report server Management for SCVMM to your **Environments** deployment, you can configure Changing reporting **Existing Lab** resources for your Management team project **Configurations** collections and the projects in those **Open the Team** collections. Foundation
- Add virtual machines as testing resources: You can configure Visual Studio Lab Management to support developing and testing your code.

<u>Console</u> <u>Review Server</u> <u>Status and</u> <u>Settings</u>

Administration

<u>View Installed</u> <u>Updates on</u> Your Server

 Monitor the servers and resources in the deployment: After you install and initially configure TFS, you can

Stop and Start Services, Application Pools, and Websites

Backing up

and Restoring monitor and change the Your configuration to support your development projects.

**Deployment Move Team** Foundation Server from

- Back up your data: One Hardware Configuration You can create maintenance plans to Another to help protect Move Team your deployment.
- Change hardware or environment: You can move your deployment to different hardware resources, or change its workgroup or domain.

Foundation Server from One **Environment** to Another

Organize and manage related projects: You can organize all the **Organizing** Your Server development projects around a specific code with Team base and the resources **Project** that support them into a **Collections** team project collection. You can create as many Create a Team team project collections Project **Collection** as you need to support development efforts.

	Configuring Resources to
Add resources to	<u>Support Team</u>
existing team projects.	Projects
You can add or change	Configure a
the resources that are	Default
available to team	Location for
project members, such	Team Project
as a team project portal.	Portals
e-mail notification of	
events. or a shared	Support alerts
document library.	and feedback
	<u>requests with</u>
	an SMTP
	<u>server</u>
	_
	Team
	Foundation
	Server
	Services and
	<u>Service</u>
	Accounts
	<u>Service</u>
	Accounts and
	<u>Dependencies</u>
	<u>in Team</u>
	<b>Foundation</b>
Manage the services	<u>Server</u>
and service accounts:	
You can start and stop	Team
services as needed for	Foundation
operational	Background
maintenance. You can	<u>Job Agent</u>
also change the	Change the
accounts or passwords	<u>Change the</u>

that are used as serviceServiceaccounts in TFS.Account or

Password for SQL Server Reporting Services

Change the Service Account or Password for Team Foundation Server

## **See Also**

#### Tasks

Add Users to Team Projects

#### Concepts

Planning and Tracking Projects <u>Configuring and Managing Your Build System</u> Planning and Tracking Projects <u>Backing up and Restoring Your Deployment</u> <u>Technical Reference for Team Foundation</u>

#### **Other Resources**

#### Managing Permissions

Managing Team Foundation Version Control

Send <u>feedback</u> on this topic to Microsoft.

- -
- $\Box$  Visual Basic  $\Box$  C#
- □ Visual C++
- □ F#
- $\Box$  HLSL
- □ JScript

Configuring Your Server Using the Team Foundation Administration Console See Also Send Feedback

You can perform many of the configuration and maintenance tasks for your deployment of Visual Studio Team Foundation Server (TFS) by using the administration console for Team Foundation. This console is installed by default on any server on which you install a component of Team Foundation. If all of your components are installed on a single server, the administration console provides management nodes for all components in your deployment. If, however, your deployment utilizes multiple servers, you must open the console on the server that is running the component that you want to manage.

## **Common Tasks**

Title

#### Description

Add and manage SharePoint Web applications: You can support your team projects with resources in SharePoint Products by installing the Team Foundation Server extensions for SharePoint Products on one or more servers and adding SharePoint Web applications as resources available for team projects and team project collections.

<u>Add a</u> <u>SharePoint</u> <u>Web</u> <u>Application to</u> <u>Your</u> <u>Deployment</u>

Modify or Remove Access Between a SharePoint Web Application and Team Foundation Server

Add and manage a reporting resource: You can support your team projects with reports and reporting abilities by adding a server that is running SQL Server Reporting Services. After you add a report server to your deployment, you can configure reporting

Add reports to a team project

resources for your team project collections and the projects in those collections.

Add virtual machines as development and testing resources: You can configure Visual Studio Lab Management to support for Your Team the use of virtual machines when developing, deploying, and testing your application.

Lab Management for SCVMM **Environments** How to: Change the Host Groups **Project** Collections

**Configuring** 

How to: Change the **Library Share** for Your Team **Project Collections** 

**Organizing** Your Server with Team **Project** Collections

Organize and manage projects and their

Set Administrator **Permissions** for Team **Project** Collections
resources: You can manage your team projects more efficiently by grouping them together and assigning the same resources to them.	Add Users to Team Projects Change access levels Collaborate (dig deeper) Collaborate Configure Team Foundation Server to Support Your Development Teams
Manage Server-Level Users and Groups: You can use the default groups and permissions at the server level to manage users and their permissions in your deployment, or you can create your own custom groups.	Set Administrator Permissions for Team Foundation Server

Monitor the servers and resources in the resources in the<br/>deployment: After you<br/>install and initially<br/>configure Visual StudioOpen the ream<br/>FoundationConsole

Server, you canReview Servermonitor and change theStatus andconfiguration toSettingssupport yourevelopment projects.

### **See Also**

#### Tasks

Create a Team Project Collection

#### Concepts

Organizing Your Server with Team Project Collections Backing up and Restoring Your Deployment Configuring Users, Groups, and Permissions

#### **Other Resources**

<u>Configuring Resources to Support Team Projects</u> <u>Get Up and Running With a Single-Server Installation [Tutorial]</u> <u>Restoring a Single-Server Deployment to New Hardware [Tutorial]</u>

Visual Basic C#
Visual C++
F#
HLSL
JScript
Open the Team Foundation Administration Console
See Also Send Feedback

You can configure and manage various aspects of your deployment of Team Foundation by using the administration console for Team Foundation. For example, you can add a server for hosting team project portals, create and modify team project collections, and change the service account for Visual Studio Team Foundation Server. This administration console is installed when you install one or more of the following components on a server:

- an application-tier for Team Foundation Server
- the extensions for SharePoint Products
- Team Foundation Build
- Visual Studio Lab Management

For more information, see The Team Foundation Administration Console.

**Required Permissions** 

You must be a member of the local Administrators group on the server where you want to open the console, and either a member of the Team Foundation Administrators group or your Edit Server-Level Information permission must be set to Allow. For more information about permissions, see <u>Team Foundation</u> <u>Server Permissions</u>.

#### To open the administration console from the Start menu

1. On a server that is running one or more components of Team Foundation (as listed earlier in this topic), choose Start, All Programs, and then choose

Microsoft Team Foundation Server.

2. Choose Team Foundation Server Administration Console.

#### ⊠Note

If Team Foundation Server Administration Console does not appear as a menu option, you might not have permission to open it. You can try to open it at a command prompt, but you might not be able to access some or all of the administration console's functionality.

The administration console opens. You might have to wait briefly for all the information to populate throughout the nodes of the console.

#### To open the administration console at a command prompt

1. On a server that is running one or more components of Team Foundation (as listed earlier in this topic), open a Command Prompt window, and change directories to the tools directory for Team Foundation Server.

By default, the tools directory is located in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools.

2. Type **TFSMgmt.exe**, and then press ENTER.

The administration console opens. You might have to wait briefly for all the information to populate throughout the nodes of the administration console.

Note You can open the administration console at a command prompt even if you do not have all the required permissions to view or modify some or all of the information that the console shows.

### See Also

#### Concepts

<u>The Team Foundation Administration Console</u> <u>Team Foundation Server Architecture</u> <u>Organizing Your Server with Team Project Collections</u>

Visual Basic C#
Visual C++
F#
HLSL
JScript
Review Server Status and Settings
See Also Send Feedback

You can use the administration console for Team Foundation to review the current configuration settings that your deployment uses. You can also use the console to review the status of the connections between Visual Studio Team Foundation Server and the components with which it is integrated, such as SQL Server Reporting Services and SharePoint Products.

Team Foundation Server creates a log file every time that the configuration of the deployment is changed. You can review these files to better understand the history of changes or to help you troubleshoot problems.

#### To review server status

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Under the server name, choose Application Tier.
- 3. Review the status of the server components of Team Foundation.

#### To review log files

- 1. Open the administration console for Team Foundation.
- 2. Under the server name, choose Logs.
- 3. Find the log that you want to review in the list, and then either double-click it to open it, or highlight it and then choose Enter.

As an alternative, choose Open Logs Folder to open the folder that contains all of the log files.

### **See Also**

#### Tasks

Add a SharePoint Web Application to Your Deployment Modify or Remove Access Between a SharePoint Web Application and Team Foundation Server

#### Concepts

Organizing Your Server with Team Project Collections

#### **Other Resources**

Configure or Reconfigure Lab Management Configuring Resources to Support Team Projects

Visual Basic C#
Visual C++
F#
HLSL
JScript
Add a SharePoint Web Application to Your Deployment
See Also Send Feedback

To integrate team projects with SharePoint sites, you must add one or more SharePoint Web applications to standard and advanced installations of Visual Studio Team Foundation Server. However, you cannot add SharePoint Web applications to basic installations of Team Foundation Server. If you add SharePoint Web applications, members of your team can perform the following tasks:

- Automatically create and configure a team project portal when they create a team project or a team project collection.
- Create a shared documents folder for projects in Team Explorer.
- Create, store, and share process guidance for your project based on the process template for that project.

To add a web application, you must configure settings on the server that hosts the web application and the server that hosts the application tier for Team Foundation. In addition, you must add the service account that SharePoint Products uses to the SharePoint Web Application Services group in Team Foundation Server. You perform all these tasks in the administration console for Team Foundation.

To integrate Team Foundation Server and SharePoint Products, you must ensure that all the following conditions are true:

- Team Foundation Server must be configured with the location of each SharePoint web application that you want to use in the deployment.
- The server that hosts the SharePoint web application must be configured with the location of Team Foundation Server.

- The service account for each SharePoint web application that you want to use must be added to the appropriate permissions group in Team Foundation Server.
- The service account for Team Foundation Server (TFSService) must have the permissions that it requires to create sites or site collections within the SharePoint Web application.

Depending on your business needs, you might want to grant access for more than one SharePoint Web application to support your deployment of Team Foundation Server. For example, you can grant access to a SharePoint Web application that supports projects that you create in the current version of Team Foundation Server. You can then create a redirection to another Web application that supports projects that you created in the previous version of Team Foundation Server.

#### **Required Permissions**

To perform these procedures, you must be a member of the following groups or have the following permissions:

- The Team Foundation Administrators group or your View instance-level information and Edit instance-level information permissions must be set to Allow.
- The Administrators group on the server that is running the application tier.
- The Administrators group on the server where the Team Foundation Server Extensions for SharePoint Products have been installed. The server on which the extensions are installed is hosting the SharePoint web application for which you want to grant access.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

# To grant access between a SharePoint web application and Team Foundation Server

1. On the server where you have installed the Team Foundation Server

Extensions for SharePoint Products, open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the tree, and choose Extensions for SharePoint Products.
- 3. In the Extensions pane, choose Grant access.

The Access for Team Foundation Server window opens.

4. In URL for Team Foundation Server, the public URL for Team Foundation Server.

You can look up this information in the Application Tier node of the administration console where you have installed the application tier for Team Foundation Server.

5. In SharePoint Web Application, choose the URL for the SharePoint web application that you want to use from the drop-down list.

If you have installed the Team Foundation Server Extensions for SharePoint Products on the server that is hosting the SharePoint Web applications, all SharePoint web applications on that server appear in the list. If no web applications appear, either the extensions were not installed correctly, or no SharePoint web applications have been configured. For more information about how to install the extensions, see Configure Team Foundation Server Extensions for SharePoint Products.

6. (optional) In Restrict site creation to the following path, specify the path of the site collection where you want to create all sites that Team Foundation Server will use.

This step is recommended for most deployments, because it helps ensure that the deployment can be backed up and restored correctly.

7. In Enterprise Application Definition, specify the name of the definition that you created for Team Foundation Server.

This step is required only if you are configuring a web application that is hosted on Microsoft Office SharePoint Server 2007 or SharePoint Server 2010 and you want reports and dashboards to operate correctly. If you are configuring a web application that is hosted on SharePoint Foundation 2010, leave this field blank.

8. When you have provided the required information, choose OK.

If all values are correct, access will be granted. This process might take several moments.

# To create a redirection for a SharePoint web application to support Team Foundation Server 2008 or 2005

1. On the server where you have installed the Team Foundation Server Extensions for SharePoint Products, open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the tree, and choose Extensions for SharePoint Products.
- 3. In the Extensions pane, choose Create redirection for previous versions.

The Redirection for Report Server window opens.

- 4. In SharePoint Web Application, choose the URL for the web application that you want to use from the drop-down list.
- 5. (optional) In Relative path, specify the path for the location where you want to create and support project sites for previous versions of Team Foundation Server.

#### **▲Important**

You must specify a path that already exists. You cannot create a path from this page.

- 6. In Report Manager URL, specify the URL (including the instance name, if applicable) for Report Manager.
- 7. In Reporting Services Web Service URL, specify the URL for the web service for SQL Server Reporting Services.
- 8. When you have provided the required information, choose OK.

If all values are correct, the redirection will be created. This process might take several moments.

# To grant access between Team Foundation Server and a SharePoint web application

1. On the server where you have installed the application tier for Team Foundation Server, open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the tree, and choose SharePoint Web Applications.
- 3. In the SharePoint Web Applications pane, choose Add.

The SharePoint Web Application Settings window opens.

4. On the General tab, in Friendly Name, specify a name for this SharePoint web application.

This name will appear in the list of web applications. If you intend to use more than one SharePoint web application in your deployment, consider specifying a name that will help users distinguish this access grant from the access grants for other SharePoint web applications. You can also add a description in the Description box to help identify this access grant, but the description will not appear in the list.

5. In Web Application URL, specify the URL of the SharePoint eb application for which you want to grant access.

This name should match the name of the web application in SharePoint

Products. If the SharePoint web application is configured to use a port number that is not the standard default (80), you must also specify r the port number.

6. In Central Administration URL, specify the URL and the port number for SharePoint Central Administration.

#### ⊠Note

If you have administrative permissions on the server that is running SharePoint Products, you can view this URL and port number by opening SharePoint Central Administration. If not, you must obtain this information from an administrator for that server.

7. In Default location for team project collection sites, specify the relative path for the site collection that you want to use as the root for team project collections that use this SharePoint web application.

#### ⊠Note

By default, the location name is /sites, but you can use any named path for site collections as configured in SharePoint Central Administration. If you are deploying Team Foundation Server in an environment where another administrator manages SharePoint Products, consider contacting that administrator for guidance on which path to use.

- 8. (optional) If you want to make sure that the relative path that you provided is valid, choose Verify Path.
- 9. Choose OK.

If all the values are correct, the SharePoint Web application appears in the list of Web applications that are available for use with Team Foundation Server. This process might take several moments.

⊠Note

Team Foundation Server will not only grant access but also attempt to add the service accounts for the SharePoint web application to the appropriate group in Team Foundation. If any one of these attempts fails, the configuration information for the web application is saved, but an error message appears. You must then manually configure the settings that the message indicates.

# **Add the Service Account for the SharePoint web Application**

When you grant access between a SharePoint web application and Team Foundation Server, you must add the service account for the web application to the SharePoint Web Application Services group in Team Foundation Server.

#### ⊠Note

You do not have to perform the procedure to grant access between a SharePoint web application and Team Foundation Server if your account is a member of the Farm Administrators group. If your account is a member of that group, these settings are configured for you automatically when you perform the procedure to grant access between Team Foundation Server and the SharePoint web application.

#### To add a service account for a SharePoint web application to SharePoint Web Application Services

1. On the server where you have installed the application tier for Team Foundation Server, open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the tree, and choose SharePoint Web Applications.
- 3. In the SharePoint Web Applications pane, in the Service Accounts for SharePoint Web Applications section, choose Add Members.

The Select Users, Computers, or Groups dialog box opens.

4. In Enter the object names to select, specify the account that you want to add, and then choose OK.

### **See Also**

#### Tasks

Modify or Remove Access Between a SharePoint Web Application and Team Foundation Server

#### Concepts

<u>Team Foundation Server Permissions</u> <u>The Team Foundation Administration Console</u> <u>Organizing Your Server with Team Project Collections</u>

#### **Other Resources**

**Configuring Resources to Support Team Projects** 

Visual Basic C#
Visual C++
F#
HLSL
JScript
Modify or Remove Access Between a SharePoint Web Application and Team Foundation Server

See Also Send Feedback

After you grant access between a SharePoint web application and Visual Studio Team Foundation Server, you can modify or delete either or both access grants. Over time, you might need to modify the access grants in your deployment to reflect changes in your organization. For example, if you change your deployment to use HTTPS and Secure Socket Layer (SSL) instead of HTTP, you must update the access grant for the web application to reflect these changes.

#### ⊠Note

When you remove an access grant, you should verify whether the service account for the SharePoint web application is being used by another web application in the deployment. If it is not, you should remove the account from the SharePoint Web Application Services group.

#### **Required Permissions**

To perform these procedures, you must be a member of the Team Foundation Administrators group or your Edit Server-Level Information permission must be set to Allow in Team Foundation Server. You must also be a member of the local Administrators group on each server that hosts the administration console for Team Foundation. For more information about permissions, see <u>Team</u> <u>Foundation Server Permissions</u>.

#### To modify or delete access from a SharePoint web application to Team Foundation Server

1. On the server that is running the Team Foundation Server Extensions for

SharePoint Products, open the administration console for Team Foundation, and expand the tree.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Under Team Foundation, choose Extensions for SharePoint Products.
- 3. In the Grant access to a SharePoint Web application for Team Foundation Server pane, highlight the access grant that you want to modify, and then perform one of the following sets of steps:
  - Choose Modify access, change the access grant, and then choose OK.
  - Choose Remove access, and in the confirmation dialog box, choose Yes.

# To modify or delete access from Team Foundation Server to a SharePoint web application

1. On the server where you have installed the application tier for Team Foundation Server, open the administration console for Team Foundation, and expand the tree.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Under Team Foundation, choose SharePoint Web Applications.
- 3. In the SharePoint Web Applications pane, highlight the access grant that you want to modify or delete, and perform one of the following steps:
  - Choose Change, change the access grant, and then choose OK.

#### ⊠Note

Any changes that you make to the access settings for the web application might have a significant impact on the team projects and team project collections that use this SharePoint web application. On the Dependents tab, you can review the list of collections and projects that use this web application. You should consider verifying whether each of these sites still operates as you expect after you modify the web application.

• Choose Remove. In the Remove SharePoint Web Application dialog box, choose Yes if you are sure that you want to remove access to the web application. If you are not sure, choose No.

#### **▲Important**

When you remove a web application, you will have a significant impact on any team projects and team project collections that use this SharePoint web application. If you remove access for a SharePoint web application, any team project or team project collection that uses this web application will lose whatever SharePoint libraries or team project portals that were hosted on that web application. Before you remove a web application, you should make sure that you understand what projects and collections will be affected. To review the list of collections and projects that use this web application, you highlight the access grant, choose Change, and choose the Dependents tab. Before you remove access for the web application, you should consider verifying whether any of these sites is still needed.

# **Remove the Service Account for the SharePoint Web Application**

When you remove a SharePoint web application or change the service account that a web application uses, you might still need to manually remove its service account. If a service account is no longer used by any SharePoint web application in the deployment, you should remove that service account from the SharePoint Web Application Services group in Team Foundation Server.

#### To remove a service account for a SharePoint web application from SharePoint Web Application Services

1. On the server where you have installed the application tier for Team Foundation Server, open the administration console for Team Foundation, and expand the tree.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Under Team Foundation, choose SharePoint Web Applications.
- 3. In the SharePoint Web Applications pane, in the Service Accounts for SharePoint Web Applications section, highlight the service account that you want to remove, and choose Remove.

#### **▲Important**

You will not be asked to verify the removal of the service account. Make sure that you have clicked the service account that you intend to remove before you choose Remove. If you remove the wrong service account, the portals that use the SharePoint web application might stop functioning correctly. You can re-add the service account at any time.

### **See Also**

#### Tasks

Add a SharePoint Web Application to Your Deployment Open the Team Foundation Administration Console

#### Concepts

<u>Team Foundation Server Permissions</u> <u>The Team Foundation Administration Console</u>

#### **Other Resources**

**Configuring Resources to Support Team Projects** 

Visual Basic C#
Visual C++
F#
HLSL
JScript
View Installed Updates on Your Server
See Also Send Feedback

To determine what updates have been installed as part of your deployment of Team Foundation Server, you can use the administration console for Team Foundation or you can view the version number of a particular file. The administration console provides more information, but might not be available in certain scenarios, such as during an upgrade from an earlier version of Team Foundation Server. In such cases, you can check the version by viewing the file version number of the Microsoft.TeamFoundation.Admin.dll file.

#### To view installed updates from the administration console

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Under the server name, choose Application Tier.
- 3. In the Data Tier Summary section, choose Installed Updates.

The Team Foundation Server Installed Updates window opens.

- 4. Review the information in the Name, KB Article, Release, and Install Date columns.
- 5. When you have finished viewing the installed updates, choose OK.

## Viewing the file version to determine updates

In some circumstances, you might need to determine the update level of your deployment without accessing the administration console. For example, the Installed Updates link in the administration console is not active until after you configure Team Foundation Server. Therefore, you must check the file version of Microsoft.TeamFoundation.Admin.dll to determine what updates are already installed.

#### To view the file version of Microsoft.TeamFoundation.Admin.dll

- 1. Open Windows Explorer (or File Explorer), and browse to \Program Files\Microsoft Team Foundation Server 11.0\Tools.
- 2. Expand the sub-menu for Microsoft.TeamFoundation.Admin.dll, and then choose Properties.
- 3. Choose the Details tab, and review the information under File Version.
- 4. When you have finished viewing the file version, choose OK.

### See Also

#### Concepts

<u>The Team Foundation Administration Console</u> <u>Backing up and Restoring Your Deployment</u>

Visual Basic 
C#
Visual C++
F#
HLSL
JScript
Organizing Your Server with Team Project Collections
Send Feedback

You can manage your team projects more efficiently by grouping them together and assigning the same resources to them. For example, you can group projects that have similar requirements or objectives, such as all projects that relate to a particular code base. You can then manage that grouping as an autonomous resource with its own user groups, server resources, and maintenance schedule. In Team Foundation Server (TFS), you group team projects together in one or more organizational units called team project collections. A team project collection is an organizing structure that you can use to define and control a group of team projects within TFS. When you create a collection, you specify the logical and physical resources that team projects within that collection can use. All the artifacts and data that those projects use are stored in the single database of the collection.

Team project collections provide server administrators with the following advantages:

- A single database that stores all the data for every project in each collection. Administrators can back up and restore this database independently of other collections. An administrator can also detach a team project collection, back it up, and then restore it to a different deployment of TFS.
- A scalable method that administrators can use to manage the resources that development efforts require. Administrators can reassign resources to better meet the demands of the projects within a collection.

Team project collections provide project administrators with the following advantages:

• A grouping of related projects that can share reports, work items, and

process guidance, as well as a code base.

• An autonomous code base that can be built, branched, merged, and iterated according to the needs of the projects within the collection. Code dependencies outside the collection can be formally managed.

If you create multiple collections, you can store all databases for them on a single instance of SQL Server, or you can distribute the databases across one or more instances.

The following illustration shows how databases for team project collections are integrated with the logical architecture of TFS:



When you install TFS, you can create a default collection to contain all team projects, or you can delay creating a collection. However, you must create at least one collection before you can create your first team project. All projects must be created within a collection.

If you upgrade TFS from an earlier version, a default collection is created, and all existing projects are stored in that collection. After you install or upgrade, you can create more collections to best suit your organizational requirements. In addition, you can choose to selectively upgrade an individual team project collection by detaching it from one deployment, backing up its database, restoring it to a deployment running the current version of TFS, and then using the Configure Features wizard to add the features of TFS 2012 to the projects in that collection. For more information, see <u>Move a Team Project Collection</u> and Configure features after a TFS upgrade.

## ■Advantages of Creating Multiple Team Project Collections

If you create more than one collection, you can better separate the operational needs for one code base or other grouping of projects from the operational needs for another grouping. Because the data for each collection is stored in its own database, you can independently manage many aspects of each collection separately from other collections in your deployment. For example, you can stop and start each collection individually. Therefore, you can schedule maintenance operations for each collection at different times.

Because each collection has its own set of users and permissions, you can help increase your operational security by isolating different code bases in different collections. You can then add users only to the collection that contains the project or projects that pertain to that particular code base.

# Disadvantages of Creating Multiple Team Project Collections

If you create more than one collection, you increase the complexity of your deployment of TFS. You must back up and restore the database for each collection, and other management and maintenance tasks also increase in proportion to the number of collections that you have. For example, you must manage the set of users and permissions for each team project collection individually.

In addition, you should consider the following facts when you decide whether to create multiple collections:

- You cannot link work items across collections.
- You cannot branch or merge code across collections.
- You cannot create queries across collections.

You can perform all of these functions across team projects within the same collection. You should consider consolidating team efforts to projects within a single collection if your development efforts will benefit from the ability to branch and merge code or you must query the status of work items that relate to the same code.

### -Common Tasks

**Topics** 

Description

Create a Team Project Collection

Configuring Resources to **Projects** 

Organize resources for supporting team Support Team projects: You can create one or more team project collections to organize and support related development projects.

Collaborate (dig deeper)

access levels

<u>Change</u>

Collaborate

#### Add a Report

Server to a Add resources to Team Project existing team project collections: You can add Collection a SharePoint Web Add a application or a server **SharePoint** that is running SQL Web Server Reporting **Application** Services to a team <u>to a Team</u> project collection after **Project** you create it. Collection

Detach the Collection	Prepare a team project collection database for restoration to a different deployment of TFS: By detaching a team project collection, you stop all jobs and save the configuration data needed for attaching the collection to a different deployment. You'll want to do this before backing up the database in preparation for a move, a partial upgrade, or restoring an individual team project collection.
<u>Move a Team</u> <u>Project</u> <u>Collection</u>	Change the location of a team project collection: You can move a team project collection from one deployment of Team Foundation Server to another.
<u>Split a Team</u> <u>Project</u> <u>Collection</u>	Reorganize what projects are in a team project collection: You can change the organization of projects in a collection by splitting it and then deleting projects from each collection until both collections have a unique set of projects.

<u>Modify a</u> <u>Team Project</u> <u>Collection</u>	Start or stop a team project collection: You can stop a team project collection to maintain it or to update an underlying component on which Team Foundation Server depends.
Modify a Team Project Collection Add Users to Team Projects Set Administrator Permissions for Team Project Collections	Change resources for team projects within a collection: You can change the resources that are available to team projects within a collection, such as the Web application that projects in the collection use. You can also change user permissions and groups at the collection level.
<u>Delete a</u> <u>Team Project</u> <u>Collection</u>	Delete a team project collection: You can increase the resources that are available to other team project collections and simplify your deployment by deleting collections that have no active or viable

projects.

## Reference

Configuring Your Server Using the Team Foundation Administration Console

Configuring Resources to Support Team Projects

Collaborate (dig deeper)

Planning and Tracking Projects

Visual Basic 
C#
Visual C++
F#
HLSL
JScript
Create a Team Project Collection
See Also Send Feedback

You can organize your projects into groups that meet your business needs by creating team project collections. For example, you can create a collection for all projects that relate to a particular code base, or you can create a collection for each business unit in your organization. For more information about team project collections and how you can use them to organize your server, see <u>Organizing</u> <u>Your Server with Team Project Collections</u>.

Before you create a team project collection, you will need the following pieces of information:

- The name and the instance of SQL Server that you want to host the database for the collection. You might also want the name of an empty database to use for the collection.
- The SharePoint Web application, if any, that will host portals for team projects in this collection. If your user account does not have permissions to create sites on that SharePoint Web application, you will also need the name and the relative path of a site that a member of the Farm Administrators group created for you.

#### **▲Important**

You can use an existing SharePoint site or site collection instead of creating one when you create a collection. However, you must perform a more complex set of procedures when you configure your deployment. Certain extensions must be installed and configured on the server that is hosting the site, and system administration will be more complex. Unless your business infrastructure requires that you use an existing SharePoint site, you should use the default option in the Create a Team Project
Collection Wizard.

• The server that is running SQL Server Reporting Services and that will host any reports for team projects in this collection, if such a server has been configured for the deployment and you want to use it for the collection. If your user account does not have permissions to create folders on that server, you will also need the name and the relative path of a folder that an administrator created for you on that server.

#### **▲Important**

You can use an existing folder when you create a collection, but the process is more complex. Unless security restrictions in your business infrastructure prevent the creation of a folder as part of creating a collection, you should use the default option in the wizard that creates the folder as part of creating the collection.

#### **Required Permissions**

To perform these procedures, you must be a member of the following groups:

- The Team Foundation Administrators group, or you must have the Create team project collection permission set to Allow.
- The Windows SharePoint Server Site Administrators group and the SharePoint Administration group in SharePoint Products.
- The SQL Reporting Services Content Managers group and the SQL Reporting Services Project Content Managers group.
- The serveradmin or sysadmin group on the server that will host the database for the team project collection.
- If you want to create a site collection, a report folder, or a database as part of creating a team project collection, the service account that the Visual Studio Team Foundation Background Job Agent uses (also known as the service account for Team Foundation Server or TFSService) must have certain permissions. For more information, see <u>Service Accounts and</u>

Dependencies in Team Foundation Server and Team Foundation Background Job Agent.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

### To create a team project collection

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, choose Create Collection.

The Create Team Project Collection wizard opens.

4. In Name, specify a name for your collection. Under Description, you can optionally specify a brief description of your collection and its purpose. Choose Next.

#### ⊠Note

You cannot use certain characters, such as slashes, in names of collections. For more information, see <u>Naming Restrictions in Team Foundation</u>.

1. In SQL Server Instance, specify the name of the server that is running SQL Server. If you want to use a named instance to host the database for this team project collection, you must also specify the name of the instance as in the following example:

ServerName\InstanceName

- 2. In Database, perform one of the following steps:
  - If you want to create a database for this team project collection, choose

Create a new database for this collection.

#### **▲Important**

The service account that the Visual Studio Team Foundation Background Job Agent uses must have permissions to create a database on the instance of SQL Server where you want to create the database.

• If you want to use a database that already exists on the instance of SQL Server, choose Use this existing database, and specify the name of the database in the text box.

#### ⊠Note

To use this option, you must specify an empty database that exists on the instance of SQL Server and to which you have been granted write permissions.

- 3. Choose Next.
- 4. If you have configured one or more SharePoint web applications to support your deployment, in the Web application list, choose a SharePoint web application, and perform one of the following steps:
  - If you want to use the default option of creating a site collection, choose Next.

A SharePoint site collection will be created, and the name of the collection is used as the name of the sub-site of the root site that is configured in the SharePoint web application.

• If the service account for Team Foundation Server is not a member of the Farm Administrators group, you cannot create a site collection as part of this wizard. You must use an existing site collection that a member of the Farm Administrators group created for you. In that case, expand Advanced configuration, choose Specify a path to an existing SharePoint site, and specify the relative path of the site collection that was created for you. Choose Verify Path, and if the path is correct, choose Next.

#### **▲Important**

Unless your business infrastructure requires that you use an existing site collection, you should use the default option to create a site collection.

- 5. If you have configured a server that is running SQL Server Reporting Services to support your deployment, perform one of the following steps:
  - In Reports, review the information for the server and the folder that will host reports for this collection, and choose Next.
  - If your user account does not have permissions to create a folder on the server that is running Reporting Services, you must use a folder that an administrator created for you on that server. In that case, expand Advanced configuration, choose Specify a path to an existing folder, and specify the relative path of the folder that has been created for you. Choose Verify Path, and if the path is correct, choose Next.

#### **▲Important**

Unless security restrictions in your business infrastructure prevent the automatic creation of a folder as part of the wizard, you should use the default option to create a folder.

6. In Lab Management, choose Configure Lab Management, and choose Next.

#### **▲Important**

If Lab Management is disabled in the list of wizard pages, you must first configure Lab Management in the administration console for Team Foundation. For more information, see <u>Configuring Lab Management for SCVMM Environments</u>.

If you do not use Lab Management in your deployment, skip to step 12.

7. In Library Shares, choose Add and Verify, choose one or more virtual machine manager (VMM) library shares that should be used to store virtual machine images for team projects in this collection, and then choose Add.

You can also change the name of the library share to a friendly name by double-clicking it. For more information, see <u>How to: Change the Library</u> <u>Share for Your Team Project Collections</u>.

8. Select or clear the Auto Provision check box for each library share, and choose Next.

If you select this check box, Team Foundation Server automatically adds the library share to the team projects you create in this collection. For team projects that have not yet been created, the share is added when the New Team Project wizard creates the project.

#### **▲Important**

You must add at least one Library Share to team project collection to enable Lab Management.

- 9. Choose Next.
- 10. In Host Groups, choose Add and Verify, choose one or more VMM host groups that should be used to deploy virtual machines for team projects in this collection, and then choose Add.

You can also change the name of the host group to a friendly name by editing it. For more information, see <u>How to: Change the Host Groups for</u> <u>Your Team Project Collections</u>.

#### **▲Important**

To enable Lab Management, you must add at least one host group to a team project collection.

The verification process now confirms the following changes:

- The service account for Team Foundation Server has been added to each host in the host groups that you have specified so that Team Foundation Server can communicate directly with the hosts on the host group.
- The network adapter that will be used to connect to the virtual machines is specified.
- 1. If the service account for Team Foundation Server (TFSService) has not been added to the administrators group on one or more hosts in these host groups, you must provide a domain account that has administrative permissions on all hosts in the host groups that you have added. This step allows the service account for Team Foundation Server (TFSService) to communicate directly with the hosts on the host group.
- 2. If you have one or more Hyper-V hosts that have multiple network adapters and if Team Foundation Server cannot determine which adapter should be used to connect the virtual machines, the Multiple Network Adapters dialog box appears for each of these hosts. In the list, specify the network adapter that corresponds to your network location that you want to use.
- 3. Select or clear the Auto Provision check box for each host group.

If you select this check box, Team Foundation Server automatically adds the host group to the team projects in this team project collection. For team projects that have not yet been created, the host group is added when the New Team Project wizard creates the project.

11. To specify the user account to use for communication between the test agents and build agents and between the test controllers and Team Foundation Server, choose Service Account.

#### Caution

For this service account, you must specify a domain account that has

limited permissions. You must specify an account that does not have any administrative permissions and that is not used by any trusted service, including the build controller and the test controller. You should consider creating two accounts to use for this service account. For more information about these two accounts, see <u>How to: Configure the Lab Service Account</u>.

- 1. On the Service Account tab, specify the name of your account in Account Name and the password in Password.
- 2. To check whether the user account is valid, choose Test.
- 3. Choose Next.
- 12. In Review Configuration, review your data. If you want to make any changes, choose Previous until you reach the page that you want to change. If all the information is correct, choose Verify.
- 13. In Readiness Checks, review the status of the checks.

A blue underlined Error indicator appears next to any configuration that contains an error. You can choose the indicator for a detailed message about the problem. You must address all errors before you can continue.

14. After all readiness checks have passed, choose Create.

The process of creating a team project collection starts.

15. After the wizard finishes, choose Close.

# See Also

#### Tasks

Move a Team Project Collection Split a Team Project Collection

#### Concepts

Configuring Your Server Using the Team Foundation Administration Console Service Accounts and Dependencies in Team Foundation Server Change access levels

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Modify a Team Project Collection
See Also Send Feedback

You can support physical and organizational changes in your business infrastructure by modifying the settings for your team project collections. For example, you can stop a collection before you perform an operation such as backing up its data or applying a service pack to the server on which a collection resides. You can then restart the collection after you complete the operation. You can change the users who have access to the collection, their permissions, and the resources that are allocated for the collection in SharePoint Products and SQL Server Reporting Services. You can modify all settings for collections in the administration console for Team Foundation.

You can find information about a team project collection on the following tabs:

- <u>General</u>: Displays the name and the uniform resource indicator (URI) of the collection. From this tab, you can also stop or start the collection, view or edit the description of the collection, and administer the group membership and permissions for users and groups in the collection.
- <u>Status</u>: Displays the status of the collection. From this tab, you can also open a log file that provides details about the collection status.
- <u>Team Projects</u>: Displays the list of team projects that the collection contains. On this tab, you can also review any descriptions of team projects that were included.
- <u>SharePoint Site</u>: Displays the URI of the site under which Web sites will be created by default for all team projects that are created in the collection. If you create or modify a team project, you can specify a different site, but this root site will be used as the default. From this tab, you can also change the default location for creating team project portals and add or remove the

configuration.

• <u>Reports Folder</u>: Displays the path under which report folders will be created by default. These folders will store the reports for team projects that are created in this collection. If you create or modify a team project, you can specify another folder, but this root path will be used as the default. On this tab, you can also change the default location and remove the configuration.

#### **Required Permissions**

To perform these procedures, you must either have the following permissions or be a member of the following groups:

- The View Instance-Level Information and Edit Instance-Level Information permissions set to Allow or a member of the Team Foundation Administrators group
- If you want to modify settings for SQL Server Reporting Services or SharePoint Products, the service account for Team Foundation Server and that is used by the Team Foundation Background Job Agent must have appropriate permissions on the instance of SQL Server or the SharePoint Web application. For more information, see <u>Service Accounts and</u> <u>Dependencies in Team Foundation Server</u> and <u>Team Foundation</u> <u>Background Job Agent</u>.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

# **Managing General Settings**

## To stop or start a team project collection

1. Open the administration console for Team Foundation.

For more information, see

Open the Team Foundation Administration Console.

• Expand the name of the server, expand Application Tier, and choose Team Project Collections.

- In Team Project Collections, highlight the collection that you want to stop or start.
- In General, choose Stop Collection or Start Collection.

If the project collection is started, only Stop Collection appears. If the project collection is stopped, only Start Collection appears.

#### ⊠Note

If you stop or start a collection, the operation can take several minutes to complete. You might need to choose Refresh to display the change in state.

## To edit the description of a team project collection

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection whose description you want to edit.

- 4. In General, choose Edit Settings.
- 5. In Edit Team Project Collection Settings, update the text, and then choose OK.

### To manage membership in groups for team project collections

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to administer membership.
- 4. In General, choose Group Membership.
- 5. In Global Groups, perform one of the following steps:
  - To add a group, choose New, and provide the required information.
  - To add users to or remove users from a group, highlight that group, choose Properties, and either choose Add, or highlight the user or group you want to remove and choose Remove.
  - To remove a group, choose Remove. This option is only available for groups that you added.
  - To rename a collection-level group, highlight it, choose Properties, and specify a different name.

### To manage permissions for users and groups in the collection

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to administer permissions.

- 4. In General, choose Administer Security.
- 5. In Global Security, highlight the user or group for whom you want to administer permissions, and then select the appropriate Allow or Deny check boxes.

For more information, see <u>Team Foundation Server Permissions</u> and <u>Team</u> <u>Foundation Server Default Groups</u>, <u>Permissions</u>, and <u>Roles</u>.

# **Displaying Status Information**

# To display the status and the log file for a team project collection

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to view the status, and then choose the Status tab.
- 4. In Status, review the information in Most Recent Servicing Activity.
- 5. To view the log, either double-click it in the list, or highlight it and then choose Enter.

# Displaying Information About Team Projects

# To display a list of team projects in a collection

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to display a list of team projects.
- 4. In Team Projects, review the list of projects and their descriptions, if any.

# ■Managing Settings for SharePoint Sites

# To display the default root location for team project portals

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to display the default location where team project portals will be created.
- 4. In SharePoint Site, review the information in Current Default Site Location.

# To add or change a default root location for creation of team project portals

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to change the default location where team project portals will be created.
- 4. In SharePoint Site, choose Edit Default Site Location.
- 5. In Edit Default Site Location, perform one of the following steps:
  - If you want to specify a different SharePoint web application, choose an application in the SharePoint Web Application list, and then specify the relative path for an existing root site or site collection.

⊠Note

If the SharePoint Web Application list is empty, no applications have been configured for use by Team Foundation Server. For more information, see

Add a SharePoint Web Application to Your Deployment.

• If you do not want to change the web application, specify the relative path for an existing root site or site collection.

• In Full Path, review the information, and choose OK if it is correct.

### To remove the default location for team project portals

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to remove the default location.
- 4. In SharePoint Site, choose Clear Configuration.

#### **▲Important**

By removing the configuration, you are removing the default location in which the Create New Team Project Wizard will create team project portals. However, you are not removing the site itself or any team project portals that already exist under it. Project administrators can still create a team project portal, but they must specify a location for it.

# **Managing Settings for Reporting**

## To view the root folder for report subfolders

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to display information.
- 4. In Reports Folder, review the information in Current Default Folder Location.

## To add or change the root folder for report subfolders

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to add or change the default root folder.

#### **▲Important**

In this release, you cannot add a report server after the installation and initial configuration of Team Foundation Server. If you did not add a report server at that time, you cannot add a folder for storing reports.

- 4. In Reports Folder, choose Edit Default Folder Location.
- 5. In Edit Default Folder Location, specify a path or folder name in Folder path on the Reporting Services Server, and then choose OK.

#### ⊠Note

This operation will fail if you type the path of a folder that does not exist on the server and you do not have permission to create a folder on that server. You must specify an existing folder if you do not have permissions to create folders on that server.

### To remove the default location for report subfolders

- 1. Open the administration console for Team Foundation.
- 2. Expand the name of the server, expand Application Tier, and choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection for which you want to remove the default location.
- 4. In Reports Folder, choose Clear Configuration, and then choose OK.

#### **▲Important**

By removing the configuration, you are removing the reporting functionality for all existing and future team projects in this collection. This change can significantly impact the functionality of all team projects in this collection. If you are not sure whether you want to make this change, choose Cancel.

# **See Also**

#### Tasks

<u>Create a Team Project Collection</u> <u>Move a Team Project Collection</u> <u>Split a Team Project Collection</u>

#### Concepts

Team Foundation Server Architecture Team Foundation Server Permissions Understanding Administrative Concepts and Tools Change access levels

#### **Other Resources**

<u>Configuring Resources to Support Team Projects</u> <u>Team Foundation Server Services and Service Accounts</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic 
C#
Visual C++
F#
HLSL
JScript
Delete a Team Project Collection
See Also Send Feedback

You can delete team project collections that are no longer being used in your deployment of Visual Studio Team Foundation Server 2012 (TFS). By deleting a collection, you ensure that users do not accidentally connect to it and work on projects within that collection, and you help simplify your deployment. Additionally, in rare cases, creation of team project collections fails or the database becomes unusable. In those cases, deleting the faulty collection also helps you maintain your deployment.

To delete a collection, you must first detach it from TFS. However, if you detach a collection, you do not delete the database for the collection from SQL Server. To truly delete a collection, you must then manually delete the database for the collection from the SQL Server instance that hosts the collection database. Similarly, if you detach a collection, you do not delete the SharePoint site collection that hosted the team project portals for projects in that collection. To delete those portals, you must also delete the site collection that supported that collection.

#### **▲Important**

Ideally, each site collection supports only one team project collection, but it is possible for more than one collection to use the same site collection. You should verify that the site collection is not used by any other team project collection or for any other purpose before you delete the site collection.

You do not need to delete the database or the site collection, but the collection will not be truly deleted until you complete these steps. If you are deleting a collection for performance or space considerations, you should consider manually deleting the database from SQL Server and the site collection from

SharePoint Products after you detach the collection.

#### ⊠Note

You can delete a team project collection at a command prompt by using the **TFSConfig Collection** command. Unlike deleting a collection from the administration console, when you use the TFSConfig Collection tool, you permanently remove any chance of reattaching the collection database. For more information, see <u>Collection Command</u>.

#### **Required Permissions**

To perform the following procedures, you must be a member of the following groups:

- a member of the Administrators security group on the server or servers that are running the administration console for Team Foundation
- a member of the sysadmin group on the server or servers that are running the instance of SQL Server that hosts the databases for TFS
- a member of the Farm Administrators group for the farm from which you are deleting the site collection, if your deployment uses SharePoint Products

For more information, see <u>Team Foundation Server Permissions</u>.

### To delete a team project collection

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the name of the server, expand Application Tier, and then choose Team Project Collections.
- 3. In the list of collections, highlight the name of the collection that you want to delete.

4. On the General tab, choose Detach Team Project Collection.

The Detach Team Project Collection Wizard opens.

- 5. (Optional) On the Provide a servicing message for the team project collection page, in Servicing Message, specify a message for users who might try to connect to projects in this collection.
- 6. Choose Next.
- 7. On the Review settings that will be used to detach team project collection page, review the details.
- 8. If you want to change any settings, choose Previous. If they appear to be correct, choose Verify.
- 9. When all the readiness checks have completed successfully, choose Detach.
- 10. On the Monitor the team project collection detach progress page, when all processes have completed, choose Next.
- 11. (Optional) On the Review supplemental information for this team project collection page, note the location of the log file.
- 12. Choose Close.

The team project collection no longer appears in the list of collections in the administration console.

- 13. Open SQL Server Management Studio, connect to the instance of the SQL Server Database Engine that hosts the collection database, and expand the instance.
- 14. Highlight the name of the collection database (by default, TFS\_CollectionName), and then delete the database.

For more information, see the following page on the Microsoft website <u>How to: Delete a Database</u>.

15. Open SharePoint Central Administration, and delete the site collection that

supported the deleted collection.

For more information, see the following page on the Microsoft website: <u>Delete a site collection (SharePoint Foundation 2010)</u> or <u>Delete a Site</u> <u>Collection</u> (Microsoft Office SharePoint Server 2007).

16. If your deployment uses reporting, you can choose to delete the folders and reports used by projects that were in the deleted team project collection by using SQL Server Report Manager. For more information, see <u>SQL Server</u> <u>documentation for deleting report folders</u>.

# See Also

#### Tasks

Create a Team Project Collection Modify a Team Project Collection Move a Team Project Collection Split a Team Project Collection

#### Concepts

Team Foundation Server Architecture

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Add a Report Server to a Team Project Collection
See Also Send Feedback

If you add a report server to a collection, project members can use default reports to track progress on team projects in that collection. Project members can also customize reports to better meet their specific needs. By default, support for reporting is added to a collection when it is created if a server that is running SQL Server Reporting Services was configured as a report server for your deployment of Team Foundation Server (TFS). However, you might not have had a report server configured for your deployment when you created the collection, or you might have overridden the default settings and not configured reporting for the collection. In either case, you can add support for reporting to a collection at any time after a report server has been added to the deployment.

#### ⊠Note

You cannot automatically add reports and reporting capabilities to the projects that already exist in that collection simply by adding a report server to a collection. All team projects that you create after you add a report server will have reporting capabilities, but any team projects that already exist in that collection will not have reporting.

#### **Required Permissions**

To perform these procedures, you must be a member of the following groups:

- The Team Foundation Administrators group, or your Edit instance-level information permission must be set to Allow.
- The Content Managers group or the TFS Content Managers group in SQL Server Reporting Services.

• To create a report folder as part of adding a report server to a collection, the service account that the Visual Studio Team Foundation Background Job Agent (TFSService) uses must have certain permissions. For more information, see <u>Service Accounts and Dependencies in Team Foundation</u> <u>Server</u> and <u>Team Foundation Background Job Agent</u>.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

# To add a report server to a deployment of Team Foundation Server

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the name of the server, expand Application Tier, and then choose Team Project Collections.
- 3. In Team Project Collections, highlight the collection to which you want to add reporting, and then choose the Reports Folder tab.
- 4. On the Reports Folder tab, choose Edit Default Folder Location.

The Edit Default Folder Location window opens.

5. In Folder path on the Reporting Services Server, specify the relative path and folder name for reports for this collection, and then choose OK. For example, if the relative path is "tfsreports" and the collection name is "Contoso1", you would type /tfsreports/Contoso1.

#### ⊠Note

If your user account has the appropriate permissions, you do not have to create this folder beforehand. It will be created for you. If the service account does not have the permissions that are required to create a folder in the path, you must specify a folder in that path that has been created for this purpose.

6. If a warning appears advising that no such folder exists, verify that you have specified the correct folder name and relative path. If you want to create a folder in that location, choose OK. If you must change the name or the path, choose Cancel.

# **See Also**

#### Tasks

Create a Team Project Add a Report Server to Your Deployment Create a Team Project Collection Move a Team Project Collection Split a Team Project Collection

#### Concepts

Planning and Tracking Projects Configure features after a TFS upgrade

#### **Other Resources**

<u>Configuring Resources to Support Team Projects</u> Add Reports to a Team Project

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Add a SharePoint Web Application to a Team Project Collection
See Also Send Feedback

If you add one or more SharePoint web applications to a team project collection, project administrators can create team project portals for the team projects in that collection. By default, a collection supports team project portals when it is created if a Web application was configured as part of your deployment of Team Foundation Server (TFS). However, you might not have had a web application configured for your deployment when you created the collection, or you might have overridden the default settings and not configured a web application for the collection. In either case, you can add one or more Web applications to a collection at any time after a web application has been added to support the deployment.

#### ⊠Note

You cannot automatically add team project portals to the projects that already exist in a collection simply by adding a SharePoint Web application to a collection. After you add a web application, you can, in the Create a New Team Project Wizard, create a project portal for the project that you are creating. However, any team projects that already exist in that collection will not have portals automatically created for them on that web application.

#### **Required Permissions**

To perform these procedures, you must be a member of the following groups:

- The Team Foundation Administrators group, or your Edit instance-level information permission must be set to Allow.
- The Site Collection Administrators group or your Full Control permission

must be set to Allow in SharePoint Products.

• For TFS and SharePoint Products to interoperate correctly, both the service account that the Visual Studio Team Foundation Background Job Agent (TFSService) uses and the service account that the SharePoint Web application uses must have certain permissions. For more information, see <u>Service Accounts and Dependencies in Team Foundation Server</u> and <u>Team Foundation Background Job Agent</u>.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

# To add a SharePoint Web application and a default root location in which project administrators will create team project portals

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the name of the server, expand Application Tier, and then choose Team Project Collections.
- 3. In the Team Project Collections list, choose the collection from the list for which you want to add a web application and a default location where team project portals will be created.
- 4. On the SharePoint Site tab, choose Edit Default Site Location.

The Edit Default Site Location window appears.

- 5. In the SharePoint Web Application list, choose an application.
- 6. In Relative Path, specify the relative path of the location where sites and sub-sites will be created on the web application for this collection.

If the site does not exist, a SharePoint Site dialog will appear asking you to confirm creation of a site. Choose Yes.

7. In Full Path, review the information, and then choose OK if it is correct.

# **See Also**

#### Tasks

Add a SharePoint Web Application to Your Deployment Add Users to Team Projects Move a Team Project Collection Split a Team Project Collection

#### Concepts

Organizing Your Server with Team Project Collections

#### **Other Resources**

**Configuring Resources to Support Team Projects** 

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Move a Team Project Collection
See Also Send Feedback

You can move a database for a team project collection from one server that is running SQL Server to another. You can also move a team project collection from one deployment of Team Foundation Server (TFS) to another, although you must first detach it from its original collection and then back up its database before the move.

#### ⊠Note

When you move a collection, ideally the version and edition of SQL Server to which you are moving the database matches the version and edition that originally hosted that database. For example, if the collection database is hosted on a server that is running SQL Server Enterprise Edition, you should move it to a server that is also running SQL Server Enterprise Edition. You should not move it to a server that is running SQL Server Standard Edition or SQL Server Express. If you must change editions, you might need to disable data compression for the databases that you want to move before you back them up and restore them. For more information, see the Microsoft Knowledge Base article.

You might want to move a team project collection in the following circumstances:

- You want to expand your deployment of TFS by adding another instance of SQL Server to it, and you want to distribute existing collections between the instances.
- You have more than one deployment of TFS in your organization, and you want to move a collection to another deployment to better align with your

business needs.

- You want to move the collection to a remote office that has its own deployment of TFS.
- You want to incrementally upgrade your deployment by detaching an individual team project collection from a deployment of TFS running an earlier version, and then move it to a server running the current version of TFS. (In this scenario, you must also then upgrade each team project within the collection by using the Configure Features wizard. For more information, see Configure features after a TFS upgrade.

#### **▲Important**

If you are moving a team project collection to a different deployment of TFS, you must detach the collection before backing up its database. Otherwise, you will not be able to attach the collection to that deployment, even though you have restored its database to the instance of SQL Server that supports that deployment.

The steps that you must perform to move a collection will depend on the particular configuration of the deployment that hosted the collection and the deployment to which you move it. For example, if you move a collection to a deployment of TFS that uses a different SharePoint Web application, either you will also need to move the site collection database that supports the collection to that SharePoint Web application and manually reconnect the team project collection to the site collection after it has been moved, or you must add the SharePoint Web application that supports the collection to the deployment of TFS to which you have moved it. Similarly, if you move a collection to a new domain, you will need to add users from that domain to the appropriate groups at the collection level and the project level.

#### ⊠Note

The procedures in this topic describe how to move an entire team project collection. If you want to move part of a collection, see <u>Split a Team Project</u> <u>Collection</u>.

#### In this topic

To move a team project collection, you must complete the following procedures in the sequence listed:

- 1. <u>Required Permissions</u>
- 2. <u>Save Reports</u>
- 3. Delete Lab Management Resources
- 4. Unmap the Collection from Microsoft Project Server
- 5. Detach the Collection
- 6. Back Up the Collection Database
- 7. <u>Rebuild the Data Warehouse and Analysis Services</u>
- 8. Prepare to Move the Site Collection Database
- 9. Move the Site Collection Database
- 10. Restore the Collection Database
- 11. <u>Attach the Collection</u>
- 12. <u>Move Reports</u>
- 13. Reconnect Resources for the SharePoint Web Application
- 14. <u>Configure the Moved Team Project Collection</u>
- 15. Configure Lab Management Resources
- 16. <u>Configure Project Server Integration</u>
- 17. Configure Projects

**Required Permissions** 

To perform the following procedures, you must be a member of the following groups:

- a member of the Administrators security group on the server or servers that are running the administration console for Team Foundation
- a member of the sysadmin group in SQL Server on the servers and instances that host databases for TFS
- a member of the Farm Administrators group for the farm to which you are configuring a Web application and changing settings for the collection, if your deployment uses SharePoint Products.

If you are moving the collection between deployments that use SharePoint Products, it is strongly recommended that the service account for TFS be a member of the Farm Administrators group in SharePoint Products in both deployments. Otherwise, you might experience errors when you attempt to detach or attach the collection.

#### ⊠Note

You can move a team project collection without granting this membership to the service account for TFS. However, errors will appear when you attach the collection, and you will need to perform additional steps to reconnect projects with their portals. Even if your operational requirements generally restrict granting this membership to the service account, you should consider adding the service account to the Farm Administrators group for the duration of the move operation.
### **Save Reports**

You can skip this procedure if you are not using a server that is running Reporting Services as part of your deployment, or if the deployment to which you are moving the collection does not use a report server.

If the deployment from which you are moving the collection uses a report server, you must save the reports used by the collection so that you can upload them to the report server that supports the deployment to which you are moving the collection.

#### To save reports

• For information about this task, see the "Exporting and Saving Reports" section in the following topic on the Microsoft Web site:

Saving Reports.

### **Delete Lab Management Resources**

You can skip this procedure if either of the following conditions is true:

- The team project collection that you are moving is not currently configured for Visual Studio Lab Management.
- You are moving the team project collection within the same domain and intend to use the same System Center Virtual Machine Manager (SCVMM) that previously supported the collection.

If you are moving the collection to a different domain or intend to use a different System Center Virtual Machine Manager, you must delete the resources that are used by Lab Management from the collection database. These resources include virtual machines, templates, team project host groups, and team project library shares. You will need to re-create the Lab Management assets after you restore and attach the collection.

#### To delete the Lab Management resources

• For information about how to remove all group hosts, library shares, and environments from a specified team project collection, see

TFSConfig Lab /Delete Command with the /External option.

## ■Unmap the Collection From Microsoft Project Server

You can skip this procedure if your deployment of TFS is not integrated with Microsoft Project Server.

Before you move a collection, you must unmap any collections that will not continue to synchronize with the instance of Project Web Access or Project Web App (PWA) that supports the deployment of TFS. Each instance of PWA can integrate with only one deployment of TFS. Collections on different deployments of TFS cannot synchronize with the same instance of PWA.

For example, you must determine which of the following scenarios best supports your business needs if you have two collections on your deployment of TFS and they are both integrated with Project Server:

- If you intend to move both collections and continue to use the same instance of PWA, you do not need to unmap the collections. However, you must re-register them after the move by following the steps in this section.
- If you intend to move only one collection, you must first unmap one of the collections. Which collection you unmap depends on how you want to support Project Server integration with your TFS deployments.
  - You must unmap a collection if you do not intend to continue integration of it or if you are moving it to a deployment of TFS that is integrated with a different instance of PWA.
  - If you intend to continue integration with the same instance of PWA, you must unmap the other collection or collections that are integrated with that instance before you move the collection.

#### To unmap a collection

• Use the **TFSAdmin ProjectServer** /**UnmapPWAFromCollection** command with the /**force** option.

#### **▲Important**

If you use the **/force** option, you will unlink all work items from Project Server, which can have unintended consequences if this option is not required. For more information, see Map a Team Project Collection to an Instance of PWA and Change the Configuration of Your Deployment.

# **Detach the Collection**

Before you move a collection, you must first detach it from the deployment of TFS on which it is running. When you detach a collection, all jobs and services are stopped, and then the collection database is stopped. In addition, the detach process copies over the collection-specific data from the configuration database and saves it as part of the team project collection database. This configuration data is what allows the collection database to be attached to a different deployment of TFS. If that data is not present, you cannot attach the collection to any deployment of TFS except the one from which it originated.

#### ⊠Note

Detaching a collection prevents users from accessing any projects in that collection but does not delete any data. To delete the data for the collection, you must delete the collection.

#### To detach a team project collection

1. Open the administration console for Team Foundation on the server that hosts the collection that you want to move.

For more information, see

Open the Team Foundation Administration Console.

• Choose Team Project Collections, and in the list of collections, choose the collection that you want to move.

• On the General tab, choose Detach Collection.

The Detach Team Project Collection Wizard opens.

• (Optional) On the Provide a servicing message for the team project collection page, in Servicing Message, provide a message for users who might try to connect to projects in this collection.

• Choose Next.

• On the Review settings that will be used to detach team project collection page, review the details.

• If you want to changes any settings, choose Previous. If they appear to be correct, choose Verify.

• When all the readiness checks have completed successfully, choose Detach.

• On the Monitor the team project collection detach progress page, when all processes have completed, choose Next.

• (Optional) On the Review supplemental information for this team project collection page, either choose or note the location of the log file.

• Choose Close.

The team project collection no longer appears in the list of collections in the administration console.

# **Back Up the Collection Database**

After you have detached the collection, you must back up its database to move it to the other server. To perform this task, you must use the tools that are provided with SQL Server.

#### **▲Important**

If your original deployment used the Enterprise or Datacenter editions of SQL Server, and you want to move the database to a server running Standard edition, you must use a backup set that was made with SQL Server compression disabled. Unless you disable data compression, you will not be able to successfully restore Enterprise or Datacenter edition databases to a server running Standard edition. To turn off compression, follow the steps in the

Microsoft Knowledge Base article.

#### To back up a collection database

• For information about this task, see the following page on the Microsoft Web site, and make sure to choose the version of SQL Server that matches your deployment: <u>Backing Up and Restoring Databases in SQL Server</u> and <u>Create Back Up Schedule and Plan</u>.

#### **▲Important**

If your original deployment used the Enterprise or Datacenter editions of SQL Server, and you want to restore databases to a server running Standard edition, you must use a backup set that was made with SQL Server compression disabled. Unless you disable data compression, you will not be able to successfully restore Enterprise or Datacenter edition databases to a server running Standard edition. To turn off compression, follow the steps in the <u>Microsoft Knowledge Base article</u>.

# **Rebuild the Data Warehouse and Analysis Services**

You can skip this procedure if you do not have reporting configured as part of your deployment.

If the deployment from which you are moving the collection uses a report server, you must rebuild the warehouse for Team Foundation and the database for Analysis Services. You must perform this step on the server from which you removed the collection to ensure that reports and dashboards work correctly for the deployment after you remove the collection.

#### To rebuild the data warehouse and the Analysis Services database

- 1. Open the administration console for Team Foundation.
- 2. In the navigation bar, choose Reporting.
- 3. In Reporting, choose Start Rebuild.
- 4. In the Rebuild the Warehouse and Analysis Services Databases dialog box, choose OK.

#### ⊠Note

The warehouses will finish rebuilding and the data will finish repopulating after the Start Rebuild action completes. Depending on the size of your deployment and the amount of data, the entire process might take several hours to complete.

### ■**Prepare to Move the Site Collection** Database

You can skip this procedure if any condition in the following list is true:

- You are moving the collection to a deployment of TFS that utilizes the same SharePoint Products resources as the original deployment.
- You plan to add the SharePoint Web application that supports the collection to the deployment of TFS to which you are moving the collection.
- The collection you are moving does not use SharePoint Products.

To move a team project collection that uses a SharePoint Web application, you must move the site collection that supports the team project collection to the Web application that will support the collection in the deployment to which you are going to move it.

#### To prepare to move the site collection database

• For information about this task, choose one of the following links to the Microsoft Web site:

For SharePoint Foundation 2010:

• To move a site collection within the same domain, see

Back up a site collection and Restore a site collection.

• To move a site collection to a different domain, see <u>Restore a site collection</u> and <u>Move a SharePoint site from one domain to another</u>.

For Microsoft Office SharePoint Server 2007:

• To move a site collection within the same domain, see <u>Back up and restore</u> <u>site collections by using built-in tools</u>.

• To move a site collection to a different domain, see <u>Archive and retrieve</u> <u>content by using stsadm import and export operations</u>.

### **Move the Site Collection Database**

You can skip this procedure if you did not perform the steps to prepare to move the site collection database, as described earlier in this topic.

If you backed up the site collection database for the team project collection that you are moving, you must restore it to the SharePoint Web application that supports the deployment of TFS to which you are moving the collection.

#### To move the site collection database

• Click one of the following links for the appropriate topic on the Microsoft Web site.

For SharePoint Foundation 2010:

• To move a site collection to a different content database, see

Move site collections between databases and Restore a site collection.

For Microsoft Office SharePoint Server 2007:

- To move a site collection within the same domain, see <u>Back up and restore</u> <u>site collections by using built-in tools</u>.
- To move a site collection to a different domain, see <u>Archive and retrieve</u> <u>content by using stsadm import and export operations</u>.

### ■Move the Team Project Collection Database

As part of moving the collection, you must restore, copy, or otherwise move the collection database to an instance of SQL Server that is configured to support the deployment of TFS to which you want to move the collection. You can choose what method you prefer to move the database.

#### To move the team project collection database

• For more information about moving a database to another instance of SQL Server, see:

0

Use the Copy Database Wizard

• Copy Databases with Backup and Restore

## Attach the Collection

After you restore the collection database, you can attach the collection to the deployment of TFS to which you want to move it.

#### ⊠Note

Warnings will appear when you attach the collection if your deployment uses SharePoint Products and the service account for TFS is not a member of the Farm Administrators group. This behavior is expected.

#### To attach the collection

- 1. Open the administration console for Team Foundation on the server that hosts the application tier for the deployment to which you want to move the collection.
- 2. Choose Team Project Collections, and then choose Attach Collection.

The Attach Team Project Collection Wizard opens.

- 3. On the Select the team project collection database to attach page, in SQL Server Instance, provide the name of the server and the instance that hosts the collection database, if it is not already listed.
- 4. In the Databases list, choose the collection database that you want to attach, and then choose Next.
- 5. On the Enter the team project collection information page, provide a name for the collection in Name if one is not already present.
- 6. (Optional) In Description, provide a description of the collection.
- 7. Choose Next.
- 8. On the Review settings that will be used to attach the team project

collection page, review the information.

- 9. If you must change any settings, choose Previous. If all the settings are correct, choose Verify.
- 10. When all the readiness checks have completed successfully, choose Attach.
- 11. On the Monitor the team project collection attach progress page, when all processes have completed, choose Next.
- 12. (Optional) On the Review supplemental information for this team project collection page, choose or note the location of the log file.
- 13. Choose Close.

The team project collection appears in the list of collections in the administration console. The SharePoint Web application that supported this collection in its original deployment will appear in the list of Web applications.

14. (Optional) Modify or remove the Web application from the list of Web applications. For more information, see

Modify or Remove Access Between a SharePoint Web Application and Team Foundation Server.

### **Move Reports**

You can skip this procedure if you are not using a report server as part of your deployment or if the deployment of TFS to which you are moving the collection uses the same report server.

You must move the reports that this collection uses to the report server for the deployment of TFS to which you have moved the collection. You must upload each report manually, which can be a lengthy process. Consider whether you want to upload all reports or whether a subset of reports will meet your business needs. You do not have to upload all reports or any reports, but only those reports that you upload will be available after the move process is completed.

#### To move reports

1. Upload each report that you want to move to the appropriate folder in Report Manager.

For more information, see the following topic on the Microsoft Web site:

#### <u>Uploading Files to a Folder</u>.

• In Report Manager, edit each report to change the data source to the new report server.

For more information, see the following topic on the Microsoft Web site: <u>How</u> <u>to: Configure Data Source Properties for a Report</u>.

• After you have uploaded and edited each report, reconnect the SharePoint Web application that supports the portals for the team projects that use the reports.

For more information, see the next procedure.

# Reconnect Resources for the SharePoint Web Application

If you are moving the collection to a deployment of TFS that does not use SharePoint Products or a report server, skip the following procedure.

You must repair the connection to the SharePoint Web application even if you move the collection to a deployment of TFS that uses the same SharePoint Web application as the deployment of TFS from which you moved the collection. You must perform this procedure to ensure that all connections are correctly set between the restored collection and the Web application.

#### To repair the connection to a SharePoint Web application

- 1. Open the administration console for Team Foundation on the server that hosts the application tier for the deployment to which you want to move the collection.
- 2. Choose SharePoint Web Applications, and in the list of Web applications, choose the Web application that will support the collection that you just attached.

The Repair Connection button appears after you select a Web application in the list.

- 3. Choose Repair Connection, and in the Repair the connection to a SharePoint Web Application dialog box, choose Repair.
- 4. When the Status window reports Reconnect operation succeeded, choose Close.

## ■Configure the Moved Team Project Collection

You can skip this procedure if you moved the collection in the same domain, intend to use the same Web application that previously supported the collection, and you want to allow access for the administrators of that collection to this deployment of TFS.

After you have moved a collection, you must update the Web application and permission groups for that collection with the appropriate settings.

#### To configure the moved collection

• For information about this task, see

<u>Modify a Team Project Collection</u> and <u>Set Administrator Permissions for Team</u> <u>Project Collections</u>.

## **■Configure Lab Management Resources**

You can skip this procedure if you moved the team project collection in the same domain and intend to use the same System Center Virtual Machine Manager (SCVMM) that previously supported the collection.

If you moved the collection to a different domain or intend to use a different System Center Virtual Machine Manager, you must recreate team project host groups and team project library shares in TFS and the virtual machines, templates, and environments in Microsoft Test Manager.

#### **To configure Lab Management resources**

1. Configure the application tier for Team Foundation.

For more information, see

Configuring Lab Management for SCVMM Environments.

• Recreate the golden master virtual machines and templates in the new SCVMM and import virtual machines and templates into the team project collection.

For more information, see <u>How to: Create and Store Virtual Machines and</u> <u>Templates Ready for Lab Management</u>.

• Recreate the environments for each team project.

For more information, see Creating an SCVMM Environment Using Stored Virtual Machines and Templates.

# **Configure Project Server Integration**

You can skip this procedure if the team project collection that you are moving is not integrated with Microsoft Project Server.

If you moved a collection and intend to continue integration between it and Project Server, you must perform the following tasks:

- Register the instance of Project Web Access or Project Web App (PWA) that supports an enterprise project plan with the application-tier server that hosts the moved team project collection.
- Wait for the metadata to synchronize.
- Unregister the application-tier server that formerly hosted the collection, if it does not host other collections that are integrated with Project Server.

#### To integrate the moved collection with Project Server

• To register an instance of PWA with the deployment of TFS that hosts the moved collection, use the **TFSAdmin ProjectServer** /**RegisterPWA** command with the /**tfs**, /**force**, and /**pwa** options. For more information, see Register an Instance of PWA to Team Foundation Server.

After you have registered the instance, you must wait for the metadata to synchronize. This process happens automatically on a predetermined schedule. For more information, see Overview of the Synchronization Process for Team Foundation Server and Project Server Integration.

 After synchronization is complete, remove the registered association between the previous instance of PWA, if any, and TFS by using the **TFSAdmin ProjectServer /UnregisterPWA** command with the /tfs and /pwa options. For more information, see Register an Instance of PWA to Team Foundation Server and Remove a Component from Participating in Data Synchronization.

### **Configure Projects**

You can skip this procedure if you moved the collection in the same domain and you want to allow access for the users of projects in that collection to this deployment of TFS.

After you configure administrators for the moved collection, either you or those administrators must add users and groups to the projects in that collection. Depending on your deployment, you might also need to configure permissions for those users in SharePoint Products and Reporting Services.

#### To add users to team projects

• For information about this task, see

#### Add Users to Team Projects.

#### To add resources to moved team projects

• For more information about this task, see <u>Configuring Resources to Support</u> <u>Team Projects</u>.

### **See Also**

#### Tasks

<u>Split a Team Project Collection</u> Connect Microsoft Excel or Microsoft Project to a team project

#### Concepts

Team Foundation Server Architecture The Team Foundation Administration Console Change the Configuration of Your Deployment Collaborate (dig deeper) Collaborate Send <u>feedback</u> on this topic to Microsoft. Visual Basic 
C#
Visual C++
F#
HLSL
JScript
Split a Team Project Collection
See Also Send Feedback

You can change the size or composition of a team project collection by splitting it into multiple collections, but this process requires that you perform many steps. These steps include creating a copy of the collection database by backing up the original collection database, restoring it using a different name, and attaching that renamed database to Team Foundation Server (TFS). After you restore the renamed collection database, you will have two functional team project collections in TFS, but those two collections will still share the same reporting folder and site collection. Reporting and team project portals will continue to function, but they will not be separated from the original team project collection. You must decide whether to move any reports that are associated with the new team project collection to a separate folder, to split the site collection that hosts team project portals, or both. These changes require extra steps, but they help ensure that each collection remains a discrete unit.

You might want to split a team project collection in the following circumstances:

- You want the projects in a collection to align with business units in your organization, and the projects in the collection are now owned by separate units.
- You upgraded from an earlier version of TFS, you have only one collection, and you want to organize your projects into separate collections for security or business alignment reasons.
- You want to change ownership of some of the projects in the collection to a remote office that has its own deployment of TFS. This scenario requires that you first split a collection and then move one of the resulting collections to the remote office deployment.

#### ⊠Note

The procedures in this topic support only splitting a team project collection. If you want to move a collection after you split it, see <u>Move a</u> <u>Team Project Collection</u>.

#### In this topic

To split a team project collection, you must complete the following procedures in the sequence listed:

- 1. <u>Delete Lab Management Resources</u>
- 2. Detach the Collection
- 3. Back Up the Collection Database
- 4. <u>Restore the Collection Database With A Different Name</u>
- 5. Attach the Original Collection Database
- 6. Attach the Renamed Collection Database
- 7. Delete Projects From the Split Collections
- 8. Reconnect SharePoint Web Application Resources
- 9. Split Reports
- 10. Start the Collections
- 11. Rebuild the Data Warehouse and Analysis Services
- 12. <u>Reconfigure Team Project Portals</u>
- 13. Configure Lab Management Resources
- 14. Configure Users and Groups for the Split Team Project Collections
- 15. <u>Configure Users and Groups for Projects</u>

- 16. <u>Split the Original Site Collection to Support the Split Team Project</u> <u>Collections</u>
- 17. Synchronize Integration with Microsoft Project Server

#### **Required Permissions**

To perform the following procedures, you must be a member of the following groups:

- the Administrators security group on the server or servers that are running the administration console for Team Foundation
- a member of the sysadmin group in SQL Server on the servers and instances that host databases for TFS

If your deployment uses SharePoint Products, it is strongly recommended that the service account for TFS be a member of the Farm Administrators group.

#### ⊠Note

You can split a team project collection without granting this membership to the service account for TFS. However, you will see errors when you attach the collection, and you will need to perform additional steps to reconnect projects with their portals. Even if your operational requirements generally restrict granting this membership to the service account, you should consider adding the service account to the Farm Administrators group for the duration of the split operation.

### **Delete Lab Management Resources**

You can skip this procedure if either of the following conditions is true:

- The team project collection that you are splitting is not configured for Visual Studio Lab Management
- You are splitting the team project collection within the same domain and intend to use the same System Center Virtual Machine Manager (SCVMM) that previously supported the collection.

If you are moving the collection to a different domain or intend to use a different System Center Virtual Machine Manager, you must delete, from the collection database, the resources that Lab Management uses. These resources include virtual machines, templates, team project host groups, and team project library shares. You will need to re-create the Lab Management assets after you restore and attach the collection.

#### To delete the Lab Management resources

• For information about how to remove all group hosts, library shares, and environments from a specified team project collection, see

TFSConfig Lab /Delete Command with the /External option.

# **Detach the Collection**

Before you split a collection, you must first detach it from the deployment of TFS on which it is running. When you detach a collection, all jobs and services are stopped, and then the collection database is stopped. In addition, the detach process copies over the collection-specific data from the configuration database and saves it as part of the team project collection database.

#### To detach a team project collection

1. Open the administration console for Team Foundation on the server that hosts the collection that you want to split.

For more information, see

Open the Team Foundation Administration Console.

• Choose Team Project Collections, and in the list of collections, choose the collection that you want to split.

In the example below, the administrator chooses "TFS Fabrikam Collection 2."

• On the General tab, choose Stop Collection.

The Team Project Collection Status Reason dialog box opens. Optionally, provide a reason for stopping the collection to be displayed to your users. Choose Stop, and wait for the collection to stop. When it is stopped, its status will show as Offline.

• On the General tab, choose Detach Collection.

The Detach Team Project Collection Wizard opens.

• (Optional) On the Provide a servicing message for the team project collection page, in Servicing Message, provide a message for users who might try to connect to projects in this collection.

• Choose Next.

• On the Review settings that will be used to detach team project collection page, review the details.

• If you want to changes any settings, choose Previous. If they appear to be correct, choose Verify.

• When all the readiness checks have completed successfully, choose Detach.

• On the Monitor the team project collection detach progress page, when all processes have completed, choose Next.

• (Optional) On the Review supplemental information for this team project collection page, choose or note the location of the log file.

• Choose Close.

The team project collection no longer appears in the list of collections in the administration console.

# **Back Up the Collection Database**

After you have detached the collection, you must back up its database before you can restore a copy to the server with a different name. That copy will become the database for the part of the original collection that you want to split into another collection. To perform this task, you must use the tools that are provided with SQL Server.

#### To back up a collection database

• For information about how to manually back up and restore individual databases, see the following pages on the Microsoft Web site, and make sure to choose the version of SQL Server that matches your deployment:

### Backing Up and Restoring Databases in SQL Server and Create Back Up Schedule and Plan.

#### **▲Important**

If your original deployment used the Enterprise or Datacenter editions of SQL Server, and you want to restore the database that you want to split to a server running Standard edition, you must use a backup set that was made with SQL Server compression disabled. Unless you disable data compression, you will not be able to successfully restore Enterprise or Datacenter edition databases to a server running Standard edition. To turn off compression, follow the steps in the <u>Microsoft Knowledge Base article</u>.

## **Restore the Collection Database**

As part of splitting the collection, you must restore the backup of the collection database to an instance of SQL Server that is configured to support the deployment of TFS. When you restore the database, you must give it a different name from the name of the original collection database.

#### ĭTip

The steps below give a general overview of how to restore a team project collection database in SQL Server 2012 using SQL Server Management Studio. For more information about how to manually back up and restore individual databases, see the following page on the Microsoft Web site, and make sure to choose the version of SQL Server that matches your deployment:

#### Backing Up and Restoring Databases in SQL Server.

#### To restore the collection database with a new name

- 1. Open SQL Server Management Studio and connect to the instance that hosts the database for the team project collection that you want to split.
- 2. In Object Explorer, expand Databases, open the sub-menu for the database you want to split, and then choose Tasks, choose Restore, and then choose Database.

The Restore Database window opens on the General page.

- 3. In Source, make sure that the team project collection database is chosen. In Destination, provide a name for the copy of the database. That name should match the name of the split team project collection. In Restore plan, make sure that the backup sets to restore are the ones you want to restore to. To make sure that these are valid sets, choose Verify Backup Media and then, in Select a page, choose Options.
- 4. In Restore options, leave all the check boxes blank. Make sure that Recovery state is set to RESTORE WITH RECOVERY. In Tail-Log

Backup, clear the Leave source database in the restoring state check box, and then choose OK.

#### ĭTip

If the restore operation fails with an error message indicating that the database is in use and cannot be overwritten, you might need to manually configure all the logical file names to reflect the new name for the database. In Select a page, choose Flies, choose the ellipsis button next to each file being restored, and make sure that the names of the files reflect the new name for the database, not the old one. Then try the restore operation again.

# Attach the Original Collection Database

After you have restored the database with a different name, you must reattach the original collection database to the deployment of TFS.

#### ⊠Note

If your deployment uses SharePoint Products and the service account for TFS is not a member of the Farm Administrators group, warnings will appear when you attach the collection. This behavior is expected.

#### To attach the collection

- 1. Open the administration console for Team Foundation.
- 2. Choose Team Project Collections, and then choose Attach Collection.

The Attach Team Project Collection Wizard opens.

- 3. On the Select the team project collection database to attach page, in SQL Server Instance, provide the name of the server and the instance that hosts the collection database, if it is not already listed.
- 4. In the Databases list, choose the collection database that you want to attach, and then choose Next.
- 5. On the Enter the team project collection information page, provide a name for the collection in Name if one is not already present. Since this is the original collection, you can choose to leave the name the same as it was before.
- 6. (Optional) In Description, provide a description of the collection.
- 7. Choose Next.

- 8. On the Review settings that will be used to attach the team project collection page, review the information.
- 9. If you must change any settings, choose Previous. If all the settings are correct, choose Verify.
- 10. When all the readiness checks have completed successfully, choose Attach.
- 11. On the Monitor the team project collection attach progress page, when all processes have completed, choose Next.
- 12. (Optional) On the Review supplemental information for this team project collection page, choose or note the location of the log file.
- 13. Choose Close.

The team project collection appears in the list of collections in the administration console. If the collection state is listed as Online, choose the collection from the list, and on the General tab, choose Stop Collection.

### ■Attach the Renamed Collection Database

After you attach the original collection database, you must attach the renamed collection to the deployment of TFS. When this collection is attached, it will remain stopped. You will not be able to start it until all duplicate projects have been removed.

#### ⊠Note

Warnings will appear when you attach the collection if your deployment uses SharePoint Products and the service account for TFS is not a member of the Farm Administrators group. This behavior is expected.

#### To attach the renamed collection database

- 1. Open the administration console for Team Foundation on the server that hosts the application tier for the deployment that will host the split collection.
- 2. Choose Team Project Collections, and then choose Attach Collection.

The Attach Team Project Collection Wizard opens.

- 3. On the Select the team project collection database to attach page, in SQL Server Instance, provide the name of the server and the instance that hosts the renamed collection database, if it is not already listed.
- 4. In the Databases list, choose the renamed collection database, and then choose Next.
- 5. On the Enter the team project collection information page, type a name for the renamed collection in Name that differs from the name of the original name of the collection.

- 6. (Optional)In Description, type a description of the collection.
- 7. Choose Next.
- 8. On the Review settings that will be used to attach the team project collection page, review the information.
- 9. If you must change any settings, choose Previous. If all the settings are correct, choose Verify.
- 10. When all the readiness checks have completed successfully, choose Attach.
- 11. On the Monitor the team project collection attach progress page, when all processes have completed, choose Next.

#### ⊠Note

If the collection is supported by a SharePoint Web application, a warning icon will appear for the attach status of the SharePoint Web application. Similarly, if the original collection included reporting, a warning icon will appear for the attach status for reports. This behavior is expected, and you can ignore it.

- 12. (Optional) On the Review supplemental information for this team project collection page, choose or note the location of the log file.
- 13. Choose Close.

The name of the collection appears in the list of collections in the administration console, and its status should display as Offline.

### **Delete Projects on the Split Collections**

Now that you have two copies of the collection attached to TFS, you must delete each project from either the original collection or the renamed collection so that no project remains in both collections.

#### **▲Important**

A project cannot exist in more than one collection. Until you delete all duplicated projects between the split collections, you will not be able to start the renamed collection.

#### To delete projects from the collections

- 1. Open the administration console for Team Foundation on the server that hosts the split collections.
- 2. Choose Team Project Collections, and in the list of collections, choose the original team project collection that you stopped in order to split it.
- 3. Choose the Team Projects tab.
- 4. In the list of team projects, choose a project that you want to delete from the collection, and then choose Delete.

The Delete Team Projects dialog box opens.

5. Select the Delete workspace data check box, leave the Delete external artifacts check box cleared, and then choose Delete.

If the Delete external artifacts check box is not cleared and your team project is configured to use Lab Management, the virtual machines and templates that are associated with the project will be deleted from System Center Virtual Machine Manager. They will no longer be available to the team project in the renamed collection.

- 6. When you have finished deleting the projects you do not want hosted in the original team project collection, choose the renamed team project collection from the list of collections. Then, on the Team Projects tab, delete the projects you do not want hosted on the new collection.
- 7. Repeat these steps until both collections contain a set of unique projects.

#### ⊠Note

For the time being, these collections will still use the same SharePoint site collection for their team project portals. You can configure them to use different site collections. The steps to do this are listed later in this topic.
# Reconnect Resources for the SharePoint Web Application

After you attach the renamed collection and remove all duplicate projects, you must repair the connection to the SharePoint Web application. Repairing the connection ensures that all connections are correctly set between the Web application and the original and renamed collections.

### To repair the connection to a SharePoint Web application

- 1. Open the administration console for Team Foundation on the server that hosts the application tier for the deployment to which you want to move the collection.
- 2. Choose SharePoint Web Applications, and in the list of Web applications, choose the Web application that supports the collections that you just attached.

The Repair Connection button appears after you select a Web application in the list.

- 3. Choose Repair Connection, and in the Repair the connection to a SharePoint Web Application dialog box, choose Repair.
- 4. When the Status window reports Reconnect operation succeeded, choose Close. This might take a few minutes. In addition, you might see some errors as part of this process, since the two collections are still using the same SharePoint default site location for their team project portals. This is expected behavior.

# **Split Reports**

You can skip this procedure if your deployment does not include a server that is running SQL Server Reporting Services or if you do not want to split reports into folders that reflect the separate collections.

After you delete projects, you must move the reports that the split collection uses into a different folder, and you must delete them from the original folder.

### **▲Important**

The report folders exist in both locations. Make sure that you move all reports appropriately before you delete any report folders.

### To split reports into separate folders

1. In Report Manager, move the reports that support the split collection into the appropriate folders for that collection.

For more information, see the following topic on the Microsoft Web site:

### Move Items Page.

• If your deployment utilizes a SharePoint Web application, you might need to repair the connection again after you move the reports before they will appear correctly. If reports do not appear correctly, follow the steps in the previous procedure to repair the connection.

# **Start the Team Project Collections**

After you delete projects, you must restart the renamed collection before you can configure users and groups for the split collections.

### To start a team project collection

- 1. Open the administration console for Team Foundation on the server that hosts the split collections.
- 2. Choose Team Project Collections, and in the list of collections, choose the collection that you stopped in order to split it.
- 3. On the General tab, choose Start Collection.

The collection starts.

4. Repeat step 2 for the collection that you attached with a new name.

# **Rebuild the Data Warehouse and Analysis Services**

You can skip this procedure if you do not have reporting configured as part of your deployment.

If the deployment on which you have split a collection uses a report server, you must rebuild the warehouse for Team Foundation and the database for Analysis Services. You must perform this step to ensure that reports and dashboards work correctly for the deployment after you split the collection and that no conflicts occur with other collections in the deployment.

### To rebuild the data warehouse and the Analysis Services database

- 1. Open the administration console for Team Foundation.
- 2. In the navigation bar, choose Reporting.
- 3. In Reporting, choose Start Rebuild.
- 4. In the Rebuild the Warehouse and Analysis Services Databases dialog box, choose OK.

### ⊠Note

The warehouses will continue to be rebuilt and the data will continue to be repopulated after the Start Rebuild action finishes. Depending on the size of your deployment and the amount of data, the whole process might take several hours to complete.

# **Reconfigure Team Project Portals**

You can skip this procedure if your deployment does not use SharePoint Products.

After you start both collections, you must reconfigure the team project portals for projects in each collection so that those portals reflect the correct data for those projects.

### To reconfigure team project portals

• Open Team Explorer, connect to each team project collection, and for each team project, configure the URL for the SharePoint site. For each team project, choose Settings, choose Portal Settings, and make sure that the Reports and dashboards refer to data for this team project check box is selected.

# **■Configure Lab Management Resources**

You can skip this procedure if you split the team project collection in the same domain and intend to use the same System Center Virtual Machine Manager (SCVMM) that previously supported the collection.

If you moved the collection to a different domain or intend to use a different System Center Virtual Machine Manager, you must recreate team project host groups. You must also recreate team project library shares in TFS and the virtual machines, templates, and environments in Microsoft Test Manager.

### **To configure Lab Management resources**

1. Configure the application tier for Team Foundation.

For more information, see

Configuring Lab Management for SCVMM Environments.

• Recreate the golden master virtual machines and templates in the new SCVMM and import virtual machines and templates into the team project collection.

For more information, see <u>How to: Create and Store Virtual Machines and</u> <u>Templates Ready for Lab Management</u>.

• Recreate the environments for each team project.

For more information, see Creating an SCVMM Environment Using Stored Virtual Machines and Templates.

# ■Configure Users and Groups for the Split Team Project Collections

You can skip this procedure if both split collections will remain in the same domain and you want to allow access for the administrators of the original collection to both collections.

After you have split a collection, you must update the permission groups for both collections with users and groups that will administer those collections.

### To configure users and groups for the collections

• For more information, see

Set Administrator Permissions for Team Project Collections.

# ■Configure Users and Groups for Projects

You can skip this procedure if the split collections will remain in the same domain and you want to allow access for the users of projects in the original collection to both collections.

After you configure administrators for the both collections, either you or those administrators must configure access for users and groups to the projects in each collection. Depending on your deployment, you might also need to configure permissions for those users in SharePoint Products and Reporting Services.

### To configure access for users to team projects

• For more information, see

Add Users to Team Projects and Add team members.

# ■Split the Original Site Collection to Support the Split Team Project Collections

You can skip this procedure if your deployment does not use SharePoint Products or if you would rather continue to use the original site collection to support both team project collections.

You can continue to use the same site collection in SharePoint Products to support both split collections. Team projects in both collections will use the same team project portals as before. All portals are hosted on the site collection that supported the original team project collection. However, this configuration not only complicates the one-to-one relationship between a team project collection and a site collection but also makes restoring your deployment potentially more difficult. To avoid this complexity, you can split the site collection that supported the original team project collection to reflect the split that you made for the team project collections.

# To split the site collection and redirect the split team project collections to use the split site collections

- 1. For information about how to split a site collection, see the following pages on the Microsoft website:
  - For SharePoint Foundation 2010, see

Move site collections to a new database (split a content database).

• For Microsoft Office SharePoint Server 2007, see <u>Move site collections to a</u> <u>new database (split a content database)</u>.

• For Microsoft SharePoint Server 2010, see <u>Move site collections between</u> <u>databases</u>.

### ĭTip

Make sure that you configure user permissions and access to the site collections to match the user access to the team project collections, as detailed earlier in this topic.

- Configure any affected team project collection to utilize the split site collection by opening the administration console, choosing the collection from the list of the team project collections, and on the SharePoint Site tab, choosing Edit Default Site Location.
- Reconfigure the team project portals for team projects in each collection so that those portals reflect the correct data for those projects.

For more information, see Reconfigure Team Project Portals earlier in this topic.

# ■Synchronize Integration with Microsoft Project Server

You can skip this procedure if the team project collection that you are splitting is not integrated with Microsoft Project Server.

You do not need to take any steps to continue synchronizing data between the collection and Project Server after you have split the collection unless you also plan to move the collection to a different server. In that case, you must follow the steps for moving a collection after you split the collection. For more information, see

Move a Team Project Collection.

After you split your team project collection, you must wait for the metadata to update as part of the synchronization between TFS and Microsoft Project Server. After synchronization, your data should appear correctly in Project Server. For more information, see Change the Configuration of Your Deployment.

### To verify synchronization

- After you have split a collection, wait for the data to synchronize. This process happens automatically on a predetermined schedule. For more information, see Overview of the Synchronization Process for Team Foundation Server and Project Server Integration.
- If one of the split collections will no longer synchronize with Project Server, unmap it by using the **TFSAdmin ProjectServer** /**UnmapPWAFromCollection** command with the /**force** option.

For more information, see Map a Team Project Collection to an Instance of PWA, Change the Configuration of Your Deployment, and Remove a Component from Participating in Data Synchronization.

### **See Also**

### Tasks

<u>Create a Team Project Collection</u> <u>Move a Team Project Collection</u>

### Concepts

<u>Team Foundation Server Architecture</u> Collaborate (dig deeper) Collaborate

### **Other Resources**

Add a Team Project Portal <u>Configuring Resources to Support Team Projects</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Configuring Resources to Support Team Projects
See Also Send Feedback

After you create a team project collection, you can add or change resources for it and the projects that it supports. For example, you can add a SharePoint Web application to a collection after you create it. When you add a resource to a collection, all projects that are subsequently created in that collection will have access to that resource. However, you can also add a resource to a project that was created before that resource was added.

You can add or configure the following resources to support team projects:

- A SharePoint Web application and site collection to host team project portals for projects in the collection
- A SharePoint site or Web site to act as the portal for a team project
- A server that is running SQL Server Reporting Services to provide default reports and dashboards that show the progress of a team project
- Guidance for team project members about the processes to follow when working in the team project
- Web Access features, such as planning and backlog management tools
- Automated e-mails that inform project members of changes in the team project

### **Common Tasks**

Title

Description

Provide reports and dashboards for team projects: You can enable default reports that show progress on your projects, as well as develop customized reports, by adding a report server to your deployment of Visual Studio Team Foundation Server.

Add a Report Server to a Team Project Collection

Configure features after a TFS upgrade

Promote collaboration between members of a team project: You can help your team work better together by creating and configuring teams in configuring the features available to those teams.

Keep users up to date automatically: You can configure e-mail

Support alerts and feedback resources to allow team requests with project members the an SMTP choice to receive e-mail server notifications when certain events occur in Customize the a team project, such as format for TFS when a work item is assigned to them or a build completes.

Delete a team project: You can help your users find the projects they need by deleting any team projects in a team project collection that are no longer in use.

Delete a Team Project (Using the Administration Console)

### See Also

### Tasks

Create a Team Project Add Users to Team Projects

### Concepts

Planning and Tracking Projects Connect to Team Projects in Team Foundation Server <u>Configuring Your Server Using the Team Foundation Administration Console</u> <u>Configuring Users, Groups, and Permissions</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Configure a Default Location for Team Project Portals
See Also Send Feedback

You can organize and manage team project portals more easily if you define a default location in which the portals for all projects in a team project collection will be created. You can define this location when you create a collection by using the Create a Team Project Collection Wizard. For more information, see <u>Create a Team Project Collection</u>.

You can modify existing collections to specify a SharePoint site or a site collection as the default location in which to create portals. Although both options are valid, you gain several advantages if you specify a site collection instead of a site. A site collection and all its SharePoint sites is a single database within SharePoint Products, just as a team project collection is a single database in Visual Studio Team Foundation Server. If you specify a site collection as the root for all portals in a team project collection, you can back up and manage the databases for a site collection and a team project collection on the same schedule. Individual SharePoint sites are only parts of a collection database. Therefore, they are much more difficult to maintain than a whole site collection. For example, you can back up and restore a site collection as a unit in SharePoint Products, but you cannot back up and restore a single site.

### **Required Permissions**

To perform these procedures, you must be a member of the Team Foundation Administrators group or your Edit Server-Level Information permission must be set to Allow. For more information about permissions, see <u>Team Foundation</u> <u>Server Permissions</u>.

# To change the default location in which team project portals are created

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Under Team Foundation, expand the name of the server, expand Application Tier, and click Team Project Collections.
- 3. Click the team project collection that you want to change, and then click the SharePoint site tab.

#### ⊠Note

The tab will not appear if no SharePoint Web applications have been configured for use in your deployment. You must configure an application before you can configure a default location for portals. For more information, see <u>Modify or Remove Access Between a SharePoint Web</u> <u>Application and Team Foundation Server</u>.

4. Click Edit Default Site Location.

The Edit Default Site Location dialog box opens.

- 5. In the SharePoint Web Application list, click the Web application that you want to use.
- 6. In Relative Path, type the relative path of the site or site collection that you want to specify as the root site for in which team project portals for this collection will be created.

#### ⊠Note

If no site exists at the location that you specify, a dialog box appears and prompts you to confirm whether you want to create a site at that location.

All team project portals will be created as sub-sites of this location unless the administrator who creates the team project specifies a different location. 7. In Full Path, verify that the path is correct, and then click OK.

### See Also

### Concepts

Organizing Your Server with Team Project Collections

### **Other Resources**

How to: Add a Team Project Portal <u>Configuring Resources to Support Team Projects</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Change access levels
Send Feedback

If you don't see the product backlog page or can't send feedback requests, it might be because you don't have the right level of access to the features in Team Web Access. There are three levels of access: standard, full or limited. The access level you can have depends on your license as described in <u>Visual Studio</u> and <u>MSDN Licensing white paper</u>.

- Any user with a TFS client-access license (CAL) can have standard access.
- Full access is available to users who have one of these MSDN subscriptions: Visual Studio Ultimate with MSDN, Visual Studio Premium with MSDN, or Visual Studio Test Professional with MSDN. With full access, you get the more advanced agile project management tools, and you can send feedback requests.
- You can give anyone limited access. There's no license required. Limited access is helpful when you want to use TFS to collaborate with your customers or stakeholders (or anyone who's not on your team, really).

### ĭTip

For a team project upgraded with the RTM release of TFS 2012, you must enable new features-such as the product backlog page and feedback requestsusing the Configure Features wizard. See Configure features after a TFS upgrade.

If you already know what the access levels are and you just want to see how to set them, jump ahead to how to do it.

# **Standard access**

Standard access includes access to the team home page, work items, source and build pages. You can manage your team on the team home page, view and manage source code, changesets, and shelvesets on the Code page, view and manage builds on the Build page, and plan and manage your work in various views.

For example, you can create, modify, and find work items on the work items page...



... and manage those work items on the task board, where you can also plan and manage your sprint.

▶ Team Foundation Server 2012 / FabrikamFiber -			
HOME WORK SOURCE BUILD board work items			
Sprint 1 backlog items team members			
	TO DO 127 h		IN PROGRESS 2 h
⊿ Hello World Web Site 4 h	Welcome Screen	Change Background Color	About Screen
J Welcome Back 5 h	Add Welcome Back Text <b>3</b>	Display User's Name in Header <b>2</b>	

# **Full access**

With full access, you can plan and organize your work on the product backlog page and manage your work with the Kanban board.

Product Backlog					
backlog items board			<u> </u>		
NEW	APPROVED	1/4	COMMITTED	2/10	DONE
Technician can see service tickets on	Change preferences		Hello World Second Visit		Wel

You can request feedback from stakeholders or customers.

ACTIVITIES
E View backlog
View board
View work items
Request feedback
View reports
Open new instance of Visual Studio

And you get some charts on the team home page.

▶ Team Foundation Server 2012 / FabrikamFiber -	
HOME WORK SOURCE BUILD	
Product Backlog Item	ACTIVITIES
Sprint 1 Burndown	<ul> <li>View backlog</li> <li>View board</li> </ul>
January 11 - January 31	View work items
129	Request feedback
<b>LZJ</b> of 224 h	Open new instance of Visit
TEAM EAVORITES	MEMBERS (5)
	Adam Barr (Fabrikam)
	Isabelle Scemla (Fabrikar
	Julia Ilyina (Fabrikam)
	Morten Rasmussen (Fabr
	Peter Waxman (Fabrikam
	Manage all members

In addition, if you have installed <u>Visual Studio Team Foundation Server 2012</u> with <u>Update 2</u>, you also have access to the Test hub that supports Web-based test case management.

# Limited access

Users with limited access can create work items and edit the work items that they created, but they don't have access to any other pages. See Work in Limited access view or Work Item Only View. Here's what limited access looks like.



## **Setting access levels**

1. From the TFS home page (for example, http://myserver:8080/tfs), open the server administration page.



2. On the Access Levels tab, add the user or group. If you don't see this tab, you need to be added to the Team Foundation Administrators group.

Control Panel			
control panel access levels	extensions		
< Export Audit Log	Access Levels		
Limited	Name Full		
Standard (Default)	Features View My Work Items		
Full	Agile Boards		
	Backlog and Sprint Planning Tools Request and Manage Feedback		
	Access levels only apply to features in Team Web Access. For more information, see MSDN.		
	Set as default access level Add 🔇 Search		
	Display Name Add Windows user or group		
	No identities found in current Add TFS group		

# **Permissions and access levels**

Of course, none of these access levels expose information that you don't have permission to view. Make sure your users have both the permissions and the access levels they need. If they're members of a team, then they probably have the permissions to use the features provided with full and standard access.

- Do you need to manage permissions to your project? Change permission levels
- Do you want to add another team member? Add team members
- Do you need to grant permissions to people who aren't on the team? Give reviewers permissions to provide feedback

Send <u>feedback</u> on this topic to Microsoft.

- -
- □ Visual Basic □ C#
- □ Visual C++
- □ F#
- □ HLSL
- □ JScript

Support alerts and feedback requests with an SMTP server

Send Feedback

For feedback requests and alerts to work, you must configure an SMTP server for TFS.

# **Configure an SMTP server for TFS**

- 1. If you aren't an administrator of TFS, get those permissions.
- 2. Open the Team Foundation Administration Console from the Start menu. Or, at a command prompt, type **TFSMgmt.exe** (located in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools).
- 3. Open email alert settings.

Team Foundation Server Administration Con	sole
File Help	
FABRIKAMPRIME     FABRIKAMPRIME     Fabrication Tier     Gif Team Project Collections     SharePoint Web Applications	Application Tier
Reporting Lab Management Proxy Server Build Configuration Extensions for SharePoint Products Checkled Backups Logs Additional Tools and Components	<ul> <li>Application Tier Summary</li> <li>Administration Console Users</li> <li>Email Alert Settings</li> </ul>
	Enabled: False Alert Settings SMTP Server:
Visual So PreEm SMTP Server: Email From Advance	Alerts ddress: ed SMTP Settings OK Cancel

4. Enable email alerts and specify the SMTP Server and the email address to use for sending emails.

🧃 Team Foundation Server Administration Console			
File Help			
FABRIKAMPRIME     Fabrication Tier     Team Project Collections     Face Paint Web Applications	Application Tier		
Reporting	Application Tier Summary		
<ul> <li>Proxy Server</li> <li>Build Configuration</li> <li>Extensions for SharePoint Products</li> <li>Scheduled Backups</li> <li>Logs</li> <li>Additional Tools and Components</li> </ul>	Email Alert Settings		
	Enabled: False Alert Settings SMTP Server:		
Visual So Email Alert Setting	gs 🛛 🛃		
FreEm	ail Alerts		
Email From A	Address: tfs anonymouse@fabrikam.com		
Advan	Iced SMTP Settings		

### Troubleshooting tips:

- Make sure your firewall is configured to allow communications between TFS and your SMTP server.
- Your SMTP server must be configured to allow anonymous senders to send email, or you must have previously created an account specifically to use as the email account for alerts. If you specify the TFS service account (TFSService), this account must be a domain account with permission to send

email.

- If you used a system account (such as Network Service) as the TFS service account, leave the User and Password fields blank for the advanced configuration, and make sure that your SMTP server is configured to allow anonymous users to send mail. Alternately, specify an email-enabled domain account to use as the email account for alerts. If you do not, email notifications will not be sent.
- 5. To verify your configuration, open alerts management. You might need to refresh your browser to see this option if you just recently enabled an SMTP server.



Create an alert and then perform an action that will trigger the alert. You can set alerts for yourself or for a team.

This task supports Request and review feedback and Set alerts, get notified when changes occur.

# **Related content**

- You can also use the TFSConfig ConfigureMail command-line tool to configure the SMTP server.
- To customize the format of TFS alert emails, you can modify the associated XSL transform files.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Customize the format for TFS email alerts
See Also Send Feedback

You can customize the format of email notifications, or alerts, that team members subscribe to. These notifications are sent when changes occur to work items, code reviews, source control files, and builds. To customize their format, you can modify their associated XSL transform files. In addition, your team members can configure project alerts in Team Web Access (TWA) for various notifications and specify whether they want HTML or plain text as the format.

Alert format transform files are stored in the following folder on the applicationtier server:

Drive:\Program Files\Microsoft Team Foundation Server 11.0\Application Tier\TFSJobAgent\Transforms\1033

You can use an XML editor, such as Visual Studio, to change any of the transform files. The following table lists the two transform files used when email notifications are sent for work item change events:

WorkItemChangedEvent.xsl	Formatting for HTML e-mail

WorkItemChangedEvent.plaintextXsl e-mail The event service uses the .xsl and the plaintextXsl files to transform the XML data for an event into an e-mail message. Edit the .xsl file to get a different format for the email notification. For HTML-formatted email messages, the core layout for all events is stored in TeamFoundation.xsl. You should make a backup copy of any files that you want to change, and then test your changes. After you have made your changes, TFS will use the modified transform the next time it generates a notification for an event.

### ⊠Note

The content of the emails that are issued is automatically generated from the TeamFoundation.xsl file and the above WorkItemChangedEvent xsl files. Modifying the TeamFoundation.xsl file is not recommended. If you do modify the contents of this file, you must thoroughly test your modifications. Incorrect modifications of this file can result in the failure of TFS email alerts and cause you to be unable to view work items, changesets, or files in a Web browser.

### **Required Permissions**

To perform these procedures, you must be a member of the Administrators security group on the Team Foundation application-tier server.

### To modify the alert format for work item changes

1. On your application-tier server, open this folder:

Drive:\Program Files\Microsoft Team Foundation Server 11.0\Application Tier\TFSJobAgent\Transforms\1033

- 2. Open WorkItemChangedEvent.xsl in Notepad or some other text editor.
- 3. Edit the file to show the message that you want in the notification email.

### ⊠Note

You should make similar changes to the WorkItemChangedEvent.plaintextXsl for any users who have requested plaintext messages. 4. Save the WorkItemChangedEvent.xsl file.

### Related content

• <u>Team Foundation Server Event Service</u>
### See Also

#### Concepts

Set alerts, get notified when changes occur

#### **Other Resources**

Support alerts and feedback requests with an SMTP server Configuring Resources to Support Team Projects

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Delete a Team Project (Using the Administration Console)
See Also Send Feedback

You can reduce the complexity of a team project and simplify the navigation for users by deleting team projects from that team project collection that are no longer used. Before you delete a project, you should make sure that it and its associated data will not be needed. Deleting a team project will remove the data for the project from the collection database in SQL Server. If the project used reporting, you will also have the option to remove reports and reporting data when you delete the project. If the project had a team project portal, all references to that portal will be removed, but the SharePoint site or Web site that acted as the portal will not be deleted. If you want to delete the portal, you must do so manually after the project has been deleted.

Deleting a team project can be done from the administration console for Team Foundation. It can also be done from the command line, but the process is significantly different, and what will be removed is also different. For more information about the command-line process, see <u>Delete a Team Project</u> [TFSDeleteProject].

#### **Required Permissions**

To perform these procedures, you must belong to the following groups or have the following permissions:

- Team Foundation Administrators group or your View instance-level information and Edit instance-level information permissions must be set to Allow
- If you want to remove reporting for this project from SQL Server Reporting Services, the Team Foundation Content Managers group or the System Administrators group

#### **▲Important**

To perform administrative tasks such as deleting team projects, your user account requires administrative permissions, and the service account that the Team Foundation Background Job Agent uses must also have certain permissions granted to it. For more information, see <u>Service Accounts and Dependencies in Team Foundation Server</u> and <u>Team Foundation Background Job Agent</u>.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

In addition to these permissions, you might need to address the following requirements on a computer that is running Windows Server 2008 or Windows Vista:

- To follow a procedure that requires Internet Explorer, you might need to start it as an administrator by clicking Start, clicking All Programs, right-clicking Internet Explorer, and then clicking Run as administrator.
- To access Report Manager, reports, or Web sites for Reporting Services, you might need to add these sites to the list of trusted sites in Internet Explorer or start Internet Explorer as an administrator.

For more information, see this topic on the Microsoft Web site: <u>User Account</u> <u>Control</u>.

#### To delete a team project from the administration console

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the server, and then click Team Project Collections.
- 3. In Team Project Collections, click the collection that contains the project that you want to delete, and then click the Team Projects tab.

4. In Team Projects, click the project that you want to delete, and then click Delete.

The Delete Team Projects window opens.

- 5. (Optional) In Delete Team Projects, select one or both of the following check boxes.
  - To remove data for this team project from Reporting Services and Visual Studio Lab Management, select the Delete external artifacts check box.
  - To remove any workspaces that are associated with this team project, select the Delete workspace data check box.
- 6. In Delete Team Projects, click Delete.

The Delete Team Projects window closes.

7. On the Team Projects tab, review the state of the team project.

The status shows Deleted, with additional information about the state of the deletion in parentheses.

- (Processing) means that the process has started and is in progress.
- (Pending) means that the deletion process has started from a client application. The deletion might be in progress or might have failed. Because the process was started from a client application, the server cannot accurately report the status of the deletion. If a project deletion remains pending for a long time, try to delete the project again from the administration console.
- (Failed) means that the deletion process started but did not complete successfully. The log file will contain specific information about the failure. Review the information about the failure, and then try to delete the project again.
- 8. (Optional) Review information about the deletion in the log file, under the Logs section in the console.

### See Also

#### Tasks

Create a Team Project <u>Move a Team Project Collection</u> <u>Split a Team Project Collection</u>

Send <u>feedback</u> on this topic to Microsoft.

 Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 Visual Studio Application Lifecycle Management
 Configure Team Foundation Build Service to Use Team Foundation Server Proxy
 See Also Send Feedback

You can configure Team Foundation Build Service to retrieve files from Team Foundation Server Proxy by modifying a registry entry on the server that is running Team Foundation Build Service. If you modify this entry, Team Foundation Build Service at the remote site can work with version control from the proxy's cache.

#### **▲Important**

If you improperly modify your computer's registry, you can cause your computer to become unstable. If you are not familiar with the registry, you should not modify it in any way.

**Required Permissions** 

To complete this procedure, you must be a member of the Administrators group on the server that is running Team Foundation Build Service.

#### To configure Team Foundation Build Service to Use Team Foundation Server Proxy

- 1. On the computer that is running a controller for Team Foundation Build Service and also has the client for Team Foundation Server Proxy installed on it, choose Start, choose Run, type regedit, and then choose OK
- 2. In the browser pane, expand HKEY\_CURRENT USER to HKEY\_CURRENT\_USER\Software\Microsoft\VisualStudio\11.0\TeamFou

If you do not see this path, you have not installed the proxy client on this computer.

3. In the details pane, set Enabled to True, and then set URL to http://ProxyServer:Port.

ProxyServer is the name of the proxy server that you want Team Foundation Build Service to use, and Port is the number of the port on which the proxy server is listening for requests. (By default, this number is 8081.)

Name	Туре	Data
(Default)	REG_SZ	(value not set)
ab AutoConfigured	REG_SZ	True
ab Enabled	REG_SZ	True
ab LastCheckTime	REG_SZ	5247036903017963333
ab LastConfiguretime	REG_SZ	5247036903019889500
赵 Uri	REG_SZ	http://FabrikamFiberProxy:8081/VersionControlProxy

4. On the File menu, choose Exit to close Registry Editor.

Team Foundation Build will now use the proxy server that you specified.

### See Also

#### **Other Resources**

Managing Remote Connections to Team Foundation Server Proxy Setting up a Build Computer

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Team Foundation Server Services and Service Accounts
Send Feedback

Visual Studio Team Foundation Server (TFS), Team Foundation, and the products with which it integrates include several services and service accounts that run on some or all of the logical tiers and physical computers that compose a deployment. Depending on the configuration of your deployment, your security needs, and what components you have installed, you can use the same account for all of these service accounts, or you can use different accounts. For clarity, each service account is referred to separately by using placeholder names. The topics in this section describe the services that are used in a deployment of TFS, the service accounts under which those services run, and how to manage those accounts.

## In This Section

Team Foundation Background Job Agent

Describes how the Team Foundation Background Job Agent service provides a general scheduling mechanism for Web services on which TFS relies.

View Team Foundation Server Services

Describes how to view the services for TFS and the Web Service application pools that you install with TFS.

Service Accounts and Dependencies in Team Foundation Server

Lists the service accounts that TFS requires, in addition to providing guidance for how to manage them and the permissions that they require. The distribution of features in your deployment determines what services and service accounts run on which physical computers. In addition to the services and service accounts that Team Foundation uses, you might need to manage the service accounts for software programs that are configured to work with TFS, such as SharePoint Products and SQL Server.

Change the Service Account or Password for SQL Server Reporting Services

Provides step-by-step guidance for how to change the service account that is used for SQL Server Reporting Services data sources and the password for that account.

Change the Service Account or Password for Team Foundation Server

Provides step-by-step guidance for how to change the service account for TFS and the password for that account.

# Reference

Managing Server Configuration with TFSConfig

Describes how to use **TFSAdminUtil.exe** to update TFS to use new credentials for the service accounts on which TFS relies.

# **Related Sections**

Team Foundation Server Architecture

TFS Administration forum

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Team Foundation Background Job Agent
See Also Send Feedback

The Visual Studio Team Foundation Background Job Agent service provides a general scheduling mechanism for Web services and jobs for Team Foundation. This Windows service is also used to run the tasks spawned by various wizards, such as the New Team Project wizard and Create A Team Project Collection wizard. The service uses the service account for Team Foundation Server, referred to as TFSService. The service runs on any server that is running a Web service or Web application in the logical application tier for Team Foundation. To operate correctly, the service account for the Team Foundation Background Job Agent service must have the permissions required for the tasks that it performs.

Some Team Foundation services have tasks that recur at regular intervals. For example, administrators might want to schedule builds on a nightly basis. To accomplish this, build services must be able to set up an automatically scheduled event in the registration database. The Team Foundation Background Job Agent service provides a single Windows-based service to schedule repeating tasks on servers that are running Team Foundation. The service runs through the registration database, identifies all Team Foundation Server Web services that have scheduled events, and schedules these tasks.

#### **▲Important**

To perform tasks such as creating team project collections, the service account that the Team Foundation Background Job Agent uses must have certain permissions granted to it. For more information, see <u>Service Accounts and</u> <u>Dependencies in Team Foundation Server</u>.

### Instances

Only one instance of the Team Foundation Background Job Agent service should be running on any application-tier server for Team Foundation. By default, the service runs under the service account that you specified when you installed Team Foundation Server. To view the status of this service on an application-tier server, click Start, click Administrative Tools, click Services, and then browse to find the service.

### **Permissions**

The Team Foundation Background Job Agent service uses the same service account as Team Foundation Server does, TFSService. To operate correctly, this account requires the following permissions:

- Log on as a service
- Farm Administrators group for any SharePoint Web applications that Team Foundation Server uses
- TFSExecRole or both of the following for any databases that Team Foundation Server uses:
  - db\_owner
  - db\_create

# **Assumptions and Limitations**

The Team Foundation Background Job Agent service runs continuously on all application-tier servers. Administrators should not need to manually stop or start this service except during system recovery. For example, you must stop this service before you restore databases. The service should restart automatically when a server is restarted.

Administrators will not directly configure the Team Foundation Background Job Agent service. Tasks that need to be scheduled are configured directly in individual components of Team Foundation, such as Team Foundation Build. When an event is added or deleted, the service automatically reconfigures the tasks scheduled in the registration database.

The Team Foundation Background Job Agent service will log only one instance of any given error until that error is resolved and a success message is recorded in the Event Log, or until the service is manually restarted. If you want to monitor the Event Log for that error message, you must first stop and restart the service.

The Team Foundation Background Job Agent service is not designed to be an all-purpose scheduling mechanism. It is not designed to provide scheduling precision beyond day of the week, hour of the day, and minute of the day. Most administrators will not need to schedule tasks beyond this level of granularity.

### See Also

#### Tasks

<u>Change the Service Account or Password for Team Foundation Server</u> <u>Change the Service Account or Password for SQL Server Reporting Services</u>

#### Concepts

Service Accounts and Dependencies in Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
View Team Foundation Server Services
See Also Send Feedback

On a server for Team Foundation, you can view the services that Visual Studio Team Foundation Server uses and the Web Service application pools that Team Foundation Server installs. Team Foundation Server includes many services that run on the application tier. Your actual services will vary based on which features of Team Foundation you have installed.

#### ⊠Note

In a single-server deployment, you log on to the application-tier server to view all services and application pools. In a deployment that has more than one server, you must log on to the appropriate server that hosts the specific service or application for Team Foundation Server.

#### **Required Permissions**

To perform this procedure, you must be a member of the Team Foundation Administrators group and a member of the Administrators group on the server on which the administration console is installed. For more information about permissions, see <u>Team Foundation Server Permissions</u>.

#### To view services that Team Foundation Server uses

- 1. On the server for Team Foundation, click Start, point to Administrative Tools, and then click Computer Management.
- 2. In the navigation pane, expand Services and Applications, and then open Services.

3. Click the Log On As column header to sort the services by which account they run under.

For a list of possible services and the service accounts under which they might run, see <u>Service Accounts and Dependencies in Team Foundation</u> <u>Server</u>.

By default, all the services for Team Foundation Server run under the service account that you specified when you installed Team Foundation Server, referred to as TFSService. These services can include the Code Coverage Analysis Service, the Team Foundation Background Job Agent service, and the Visual Studio Team Foundation Build service.

#### ⊠Note

The services that appear in the list will depend on what components you have installed. For example, the Visual Studio Team Foundation Build Service is installed only when a build computer is installed.

# To view Web Service application pools that Team Foundation Server uses

- 1. On the server for Team Foundation, click Start, point to Administrative Tools, and then click Computer Management.
- 2. In the navigation pane, expand Services and Applications, expand Internet Information Services (IIS) Manager, and then click Application Pools.
- 3. Under Status or State, verify that Running or Started appears.
- 4. To view or correct the application pool identity, see How to: Verify or Correct the IIS Configuration for an Application Pool.

### **See Also**

#### Tasks

<u>Stop and Start Services, Application Pools, and Websites</u> <u>Change the Service Account or Password for SQL Server Reporting Services</u> <u>Change the Service Account or Password for Team Foundation Server</u>

#### Concepts

Service Accounts and Dependencies in Team Foundation Server Team Foundation Server Concepts Team Foundation Server Architecture

#### **Other Resources**

<u>Team Foundation Server Services and Service Accounts</u> How to: Verify or Correct the IIS Configuration for an Application Pool

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Service Accounts and Dependencies in Team Foundation Server
See Also Send Feedback

You can better manage Visual Studio Team Foundation Server (TFS) if you understand the services and several service accounts that every deployment of TFS includes and on which every deployment depends. Depending on how you have installed and configured TFS, these services and service accounts might all run on one computer, or they might run on many computers. This changes certain aspects of managing your deployment. For example, if the server-side components of your deployment run on more than one computer, you must ensure that the service accounts your deployment uses have the access and permissions they require to function correctly.

Team Foundation Server has services and service accounts that run on the following computers in a deployment:

- any server that hosts one or more databases for Team Foundation Server
- any server that hosts components of the application tier for Team Foundation
- any computer that is running Team Foundation Server Proxy
- any build computer
- any test machine
- any computer that is running one or more components of Visual Studio Lab Management

You can install and deploy different features of TFS in various ways. The distribution of features in your deployment determines what services and service accounts run on which physical computers. In addition, you might need to

manage the service accounts for software programs that are configured to work with TFS, such as the service accounts for SharePoint Products and SQL Server.

# ■Service Accounts for Team Foundation Server

Although TFS uses several service accounts, you can use the same domain or workgroup account for most or all of them. For example, you can use the same domain account "Contoso\Example" as both the service account for Team Foundation Server (TFSService) and the data sources account for SQL Server Reporting Services (TFSReports). However, different service accounts can require different permission levels. For example, TFSService must have the Log on as a service permission, and TFSReports must have the Allow log on locally permission. If you use the same account "Contoso\Example" for both, you must grant both of these permissions to it. In addition, TFSService requires significantly more permissions to operate correctly than those that TFSReports requires, as the table later in this topic shows. For security purposes, you should consider using separate accounts for these two service accounts.

#### **▲Important**

You must not use the account that was used to install Team Foundation Server as the account for either of these service accounts.

If you have deployed Team Foundation Server in an Active Directory domain, you should set the Account is sensitive and cannot be delegated option for service accounts. For example, in the following table, you should set that option for TFSService. For more information about required service accounts and placeholder names used in documentation for Team Foundation Server, see the topic "Accounts Required for Installation of Team Foundation Server" in the installation guide for Team Foundation. For more information about how to restrict account delegation in Active Directory, see the following page on the Microsoft website:

#### **Enabling Delegated Authentication.**

Because you must manage several service accounts, each service account is

referred to by a placeholder name that identifies its function, as listed in the table later in this topic. The placeholder name is not the actual name of the account that you use for each service account. The actual name of the account varies depending on your deployment. In the previous example, the account used for both TFSService and TFSReports was "Contoso\Example." In your own deployment, you might create domain accounts with the specific names of "TFSService" and "TFSReports," or you might use the system account Network Service as the service account for Team Foundation Server.

#### **▲Important**

Unless specifically stated otherwise, no groups or accounts in the following table should be members of the Administrators group on any of the servers in your deployment of Team Foundation Server.

The following table shows all the service accounts that you might use in a deployment of TFS:

Service Account	Placeholder name and usable account type	Required Permission and Group Membership	Notes
Service account for Team Foundation Server	TFSService, which can be a local account, a domain account, Local Service in a workgroup, or Network Service in a domain	<ul> <li>Log on as a service on the application-tier server</li> <li>Farm Administrators group for any SharePoint Web applications that Team Foundation Server uses<sup>1</sup></li> <li>TFSExecRole, or if this role does not exist for the database, a combination of the following roles for any databases that Team Foundation Server uses:</li> </ul>	This service account is us for all of the Web services for Team Foundation Server. If you use a domain account for this account, must be a member of a domain that a computers

		<ul><li> db_owner</li><li> db_create</li></ul>	throughout th deployment fully trust.
Data sources account for SQL Server Reporting Services	TFSReports, which can be a local account, a domain account, or Local Service in a workgroup	<ul> <li>Allow log on locally on the application-tier server and on the server that is running SQL Server Reporting Services</li> <li>TFSWareHouseDataReader on the report server</li> </ul>	This service account retrieves data for reports from Reporting Services.
Service account for Team Foundation Build	TFSBuild, which can be a local account, a domain account, or Local Service in a workgroup	Log on as a service	This service account is us when builds are configure and when build status information i communicate between the build controller and the build agents.
Service account for Lab Management	TFSLab, which can be a local account, a domain account, Local Service in a workgroup, or	Log on as a service	This service account is us when information about Lab Management communicate between Tear Foundation

	Network Service in a domain	Server and th lab agent that is running on virtual machine.
Service account for Team Foundation Server Proxy	TFSProxy, which can be a local account, a domain account, Local Service in a workgroup, or Network Service in a domain	This service account is us for all of the proxy service If you use a domain account for this account, must be a member of a domain that a computers throughout th deployment fully trust.
Service account for Test Agent and Test Agent Controller	TFSTest, which can be a local account, a domain account, Log on as a service or Network Service in a domain.	This service account is us when information about tests is communicate between the test agent controller and the test agent
		You must add at least one service

Service accounts for SharePoint Web applications	WebAppService	Allow log on locally	account for each SharePoint Web application that you configure for use with Teau Foundation Server. This service account is us to create tean project portal and to enable dashboard functionality.
Service account for Visual Studio Online	Account Service (CollectionName)	None. This account is only used if you are using a hosted deployment of TFS. It is automatically created for you, and can be viewed through the administration page of Team Web Access.	This service account is created automatically when you create a collection in Visual Studic Online, and i used when clients communicate with the host service.

<sup>1</sup> You can integrate your deployment with SharePoint Products without this permission, but you must perform additional steps if the service account is not a member of the Farm Administrators group. For more information, see Integrate with SharePoint Products Without Administrative Permissions.

# ■Services That Run Under Service Accounts

The following services run under service accounts in a deployment of Team Foundation Server:

Service name	Service account	Logical Tier
Code Coverage Service	TFSService	application tier
Team Foundation Server Web Services	TFSService	application tier
SQL Server Reporting Services (MSSQLSERVER or InstanceName if using a named instance)	Local System or a domain account	application tier
Report Web Service	Local System, Network Service, or a domain account	application tier

SharePoint Administration (if SharePoint Local Products is System, installed and Network application configured for use Service, or tier with Team a domain Foundation account Server) SharePoint Timer (if SharePoint Products is installed and application Domain configured for use account tier with Team Foundation Server) Visual Studio Team Foundation **Build Service** build **TFSBuild** Host (if Team computer Foundation Build is installed) Visual Studio TFSService application Team Foundation **Background Job** tier Agent Visual Studio Test TFSTest any Controller computer

Visual Studio Test Agent	TFSTest	test computer
Analysis Server (MSSQLSERVER or InstanceName if you are using a named instance)	Local System or a domain account	data tier
SQL Server Browser	Local Service or a domain account	data tier
SQL Server (MSSQLSERVER or InstanceName if using a named instance)	Local System, Network Service, or a domain account	data tier
SQL Server Agent (MSSQLSERVER or InstanceName if using a named instance)	Local System, Network Service, or a domain account	data tier
Account Service (CollectionName)	Automatic	web tier (Visual Studio Online only)

For more information about service accounts for SQL Server, see the following page on the Microsoft website:

SQL Server Books Online. For the most recent information about service accounts in Team Foundation, see Installing Team Foundation Server. ✓Note

If you change the service account for Team Foundation Build, you must make sure that the new service account is a member of the Build Services group. You must also make sure that the account has read/write permissions to the temporary folders and the ASP.NET temporary folder. Similarly, if you change the service account for the Team Foundation Server Proxy service, you must make sure that the account is a member of the appropriate groups. For more information, see Setting Up a Build Computer.

### See Also

Tasks

<u>Change the Service Account or Password for SQL Server Reporting Services</u> <u>Change the Service Account or Password for Team Foundation Server</u>

#### **Other Resources**

<u>Managing Server Configuration with TFSConfig</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Change the Service Account or Password for SQL Server Reporting Services
See Also Send Feedback

You can help improve the security of Team Foundation Server (TFS) by changing the service account that it uses for the data sources for SQL Server Reporting Services or by changing the password that is used for that account. TFS acts in the security context of a service account when it retrieves project data from the data sources in SQL Server Reporting Services. TFS documentation refers to this service account by the placeholder TFSReports. The actual account name depends on your installation. You might need to change the password of that account, or designate a different account. For example, if the password of the underlying account expires, and you assign a new password, you must change the password of the TFSReports account in TFS to match.

The **TFSConfig** utility does not create a new account to use as the data sources account, nor does the utility change the account password. Instead, the utility updates TFS to use a different set of credentials.

You change the password or account used as the TFSReports account by using the TFSConfig command-line utility with the Accounts option.

#### **▲Important**

The **TFSConfig** utility changes only those services that run under the old account.

You can use the same utility to assign a different account to be the TFSReports account, but you might need to perform one or more of the following additional actions:

- Before you assign an account to use as the TFSReports account, you must verify that it is a member of a workgroup or domain that is trusted by every computer in the deployment of Team Foundation.
- You must manually grant the account that you will use as the TFSReports account the Allow log on locally permission. The TFSConfig utility does not grant this permission when it assigns the account.
- Optionally, after you use TFSConfig to specify an account to use as the TFSReports account, you can revoke its Log on as a service permission, which TFSConfig automatically grants to the TFSReports account. TFSReports does not need this permission, but the TFSService account does. Therefore, you should not remove this permission if you use the same domain or workgroup account for both service accounts.

For more information about the Log on as a service permission, see this page on the Microsoft Web site: <u>Add the Log on as a service right to an account</u>. For more information about the Allow log on locally permission, see this page on the Microsoft Web site: <u>Allow log on locally</u>.

For more information about required service accounts, see the topic <u>Service</u> <u>Accounts and Dependencies in Team Foundation Server</u> and also the topic "Accounts Required for Installation of Team Foundation Server" in the installation guide for Team Foundation.

#### **Required Permissions**

To perform these procedures, you must be a member of the Administrators group on the server where TFSConfig is installed. You must also be a member of the sysadmin group on the server that hosts the configuration database. For more information about permissions, see <u>Team Foundation Server Permissions</u>.

In addition to these permissions, you might need to address the following requirements:

- To follow a command-line procedure, you might need to open an elevated Command Prompt by clicking Start, right-clicking Command Prompt, and clicking Run as Administrator.
- To access Report Manager, reports, or Web sites for SQL Server Reporting

Services, you might need to add these sites to the list of trusted sites in Internet Explorer or start Internet Explorer as an administrator.

For more information, see this page on the Microsoft Web site: <u>User Account</u> <u>Control</u>.

# ■Changing the Password or Account for the Reporting Services Account

To change the password of the TFSReports account or to assign a different account, you must log on to a server that hosts the application services for Team Foundation and use the **TfsConfig Accounts** utility.

#### ⊠Note

Depending on your deployment configuration, you might need to restart Internet Information Services (IIS) after you complete this procedure for the changes to take effect.

#### To change the password using the TFSConfig utility

1. Open a Command Prompt window and change to the directory that contains the **TFSConfig** utility.

By default, the utility is located in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools.

2. At the command line, type **TFSConfig Accounts /UpdatePassword** /accountType:ReportingDatasource /account:AccountName /password:newPassword, and then press ENTER.

Replace AccountName with the name of the current TFSReports account. Replace newPassword with the new password of the account.

#### To use the administration console to change the password

1. Open the administration console for Team Foundation on the server that hosts the application tier.

For more information, see **Open the Team Foundation Administration**
Console.

- 2. In the console, expand the server name and choose Application Tier.
- 3. In the Application Tier pane, navigate to Reporting Services Summary choose Update Account Password.

The Update Account Password window opens.

#### ⊠Note

If you used a system account as the service account, you will see an error message when you choose Update Account Password. You do not need to change the password of that account. System accounts do not have user-managed passwords.

4. Type the new password in Password, and then choose OK.

The Change Report Reader Account window opens.

5. Wait for all the status messages to complete in Status, and then choose Close.

#### ⊠Note

This process might take a few minutes.

## To assign a new reporting services service account to all Team Foundation Server services using the TFSConfig utility

1. Open a Command Prompt window and change to the directory that contains the **TFSConfig** utility.

By default, the utility is located in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools.

2. At the command line, type **TFSConfig Accounts /change** 

/accountType:ReportingDatasource /account:NewAccountName /password:newPassword, and then press ENTER.

Replace NewAccountName with the name of the new TFSReports account. Replace newPassword with the password of the account.

## To use the administration console to change the account

- 1. Open the administration console for Team Foundation on the server that hosts the application tier.
- 2. In the console, expand the server name and choose Application Tier.
- 3. In the Application Tier pane, navigate to Reporting Services Summary, and then choose Change Account.

The Change Report Reader Account window opens.

- 4. Perform one of the following steps:
  - 1. If you want to use a system account, choose Use a system account, and then choose a system account from the drop-down list.

#### ⊠Note

System accounts do not have user-managed passwords. If you choose to use a system account as TFSReports, you should not type a password in the password field.

- 2. If you want to use a domain or workgroup account, choose Use a user account, type the name of the account in Account Name, and then type the password for that account in Password.
- 5. Choose OK.

The Change Report Reader Account window opens.

6. Wait for all the status messages to complete in Status, and then choose Close.

# ⊠Note

This process might take a few minutes.

# See Also

Tasks

Change the Service Account or Password for Team Foundation Server

Reference

Accounts Command

Concepts

Service Accounts and Dependencies in Team Foundation Server

#### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Change the Service Account or Password for Team Foundation Server
See Also Send Feedback

You can help improve the security of Visual Studio Team Foundation Server 2012 (TFS) by changing its service account or the password used for that account. TFS runs services such as its Web services and the Team Foundation Background Job Agent in the context of a service account. For simplicity, the TFS documentation refers to this account as TFSService, although that is not the actual name of the account unless you specifically create an account with that name. TFS stores a record of the name of the actual account that is used as its service account. By changing the record, you can assign a different account to act as the service account. You can also change the password for that account. Whether you change the account, the password, or both, you stay synchronized with other components in your deployment. For example, if an Active Directory domain policy requires that all passwords expire periodically, you can update the password information for the service account in TFS when that password changes.

#### ⊠Note

TFS and its utilities cannot create a new local or domain account to use as TFSService, and they cannot update the password for that account in the workgroup or the domain. Instead, the utilities update the records to match the new credentials. If you your deployment includes more than one application-tier server, you must manually update each server with any changes to the service account or its password.

For more information about service accounts in TFS, see <u>Service Accounts and</u> <u>Dependencies in Team Foundation Server</u>. For more information about the accounts required for installation, including the service account for TFS, see

Accounts Required for Installation of Team Foundation Server.

**Required Permissions** 

To perform these procedures, you must be a member of the Administrators group on the Team Foundation application-tier server and a member of the sysadmin group on the server and instance of SQL Server that hosts the configuration database for Team Foundation. For more information, see <u>Team Foundation</u> <u>Server Architecture</u> and <u>Team Foundation Server Permissions</u>.

To follow a command-line procedure, you might need to open an elevated Command Prompt window. To perform this task, click Start, right-click Command Prompt, and click Run as Administrator. For more information, see this page on the Microsoft Web site: <u>User Account Control</u>.

# ■Changing the Password of the Service Account

To change the password of TFSService, you must log on to the application-tier server for Team Foundation and either use the administration console for Team Foundation, or open a Command Prompt window and use the TFSConfig command-line utility. If your deployment includes more than one application-tier server, you must perform this task on each server, or the account information will be out of synchronization.

#### ⊠Note

Depending on your deployment configuration, you might need to restart Internet Information Services (IIS) after you complete the procedure before the changes will take effect.

# To use the administration console to change the password

1. Open the administration console for Team Foundation on the server that hosts the application tier.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. In the console, expand the server name and choose Application Tier.
- 3. In the Application Tier pane, choose Update Account Password.

The Update Account Password window opens.

⊠Note

If you used a system account as the service account, you will see an error message when you choose Update Account Password. You do not need to

change the password of that account. System accounts do not have usermanaged passwords.

4. Type the new password in Password, and then choose OK.

The Change Service Account window opens.

5. Wait for all the status messages to complete in Status, and then choose Close.

⊠Note

This process might take a few minutes.

## To use the TFSConfig utility to change the password

1. On the application-tier server, open a Command Prompt window and change directories to the directory that contains the **TFSConfig** utility.

By default, this utility is located in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools.

- 2. At the command line, type **TFSConfig Accounts /UpdatePassword** /accountType:ApplicationTier /account:AccountName /password:NewPassword, and then choose ENTER.
- 3. You must specify both the name of the account you want to use as TFSService (AccountName) and the password of the account (NewPassword).

# ■Assigning a Different Account as the Service Account

To configure TFS to use a different account as the service account for Team Foundation, you can use either the administration console or the TFSConfig command-line utility. If your deployment includes more than one application-tier server, you must perform this task on each server, or the account information will be out of synchronization. Before you use either utility to make the change, consider the following issues:

- You must choose a new account that is either a system account or a member of a workgroup or domain that is trusted by every computer in this deployment of TFS.
- The configuration utilities grant the Log on as a service permission to the new service account, but the utilities do not revoke this permission from the account previously used as the service account if another service still uses that account. If the old account no longer needs that permission for the service for which it is still in use, you might want to manually remove that permission from the old account.

For more information, see this page on the Microsoft Web site: <u>Add the Log</u> <u>on as a service right to an account</u>.

- You might need to restart IIS after you complete the procedure before the changes will take effect.
- The TFSConfig utility changes only those services that run under the old account.

# To use the administration console to change the service account

- 1. Open the administration console for Team Foundation on the server that hosts the application tier.
- 2. In the console, expand the server name and choose Application Tier.

3. In the Application Tier pane, choose Change Account.

The Update Service Account window opens.

- 4. Perform one of the following steps:
  - 1. If you want to use a system account, choose Use a system account, and then choose a system account from the drop-down list.

If your server is a member of an Active Directory domain, the default choice for the system account to use is Network Service. If your server is a member of a workgroup, the default choice is Local Service. Depending on the details of your deployment, the default choice might be the only available choice.

#### ⊠Note

System accounts do not have user-managed passwords. If you choose to use a system account as TFSService, you should not type a password in the password field.

- 2. If you want to use a domain or workgroup account, choose Use a user account, type the name of the account in Account Name, and then type the password for that account in Password.
- 5. Choose OK.

The Change Service Account window opens.

6. Wait for all the status messages to complete in Status, and then choose Close.

#### ⊠Note

This process might take a few minutes.

### To use the TFSConfig utility to change the service account

1. On the application-tier server, open a Command Prompt window and change directories to the directory that contains the **TFSConfig** utility.

By default, this utility is located in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools.

2. At the command line, type **TFSConfig Accounts /change** /accountType:ApplicationTier /account:AccountName /password:NewPassword, and then choose ENTER.

For more information, see <u>Accounts Command</u>.

# See Also

#### Tasks

Change the Service Account or Password for SQL Server Reporting Services

#### Reference

Accounts Command

#### Concepts

Service Accounts and Dependencies in Team Foundation Server

#### **Other Resources**

Managing Server Configuration with TFSConfig How to: Change the Password for Visual Studio Team Foundation Build Service

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Stop and Start Services, Application Pools, and Websites
See Also Send Feedback

To ensure that Visual Studio Team Foundation Server functions correctly, you must verify that all required services, application pools, and Web sites are running on the appropriate server. In single-server deployments, each element must be running on the server that is running Team Foundation Server. In multiple-server deployments, each element must be running on the appropriate server. In addition, you might need to stop an element to perform a particular task, such as moving your deployment to a different set of hardware. For detailed information about how to start or stop all of these elements for operations such as backing up or restoring databases, see <u>TFSServiceControl Command</u>. For information about how to stop individual services or application pools, see the steps at the end of this topic.

#### **▲Important**

You must use the **TFSServiceControl** command to perform certain maintenance tasks on the server, such as backing up and restoring databases. Do not use the steps at the end of this topic to attempt to manually stop all of the services and application pools for tasks that require the TFSServiceControl command.

The following table lists the server on which each service, application pool, and Web site must be running. The Name column lists the display name for each element with service names in parentheses. Which services you need will vary based on which features of Team Foundation you have installed.

```
Element Location Name
```

		<ul> <li>Code Coverage Analysis Service</li> </ul>
		<ul> <li>Internet Information Services Administration Service (IISADMIN)</li> </ul>
		• HTTP SSL (HTTPFilter)
Services	Application- tier server	<ul> <li>Visual Studio Team Foundation Build (VSTFBUILD) (only when Team Foundation Build is installed)</li> </ul>
		<ul> <li>Visual Studio Team Foundation Background Job Agent (TFSJobAgent)</li> </ul>
		<ul> <li>World Wide Web Publishing Service (W3SVC)</li> </ul>
	Server that hosts the databases for Team Foundation	<ul> <li>SQL Server (TFSINSTANCE)</li> <li>SQL Server Agent (TFSINSTANCE) (SQLSERVERAGENT)</li> </ul>
	Server that	<ul> <li>IIS Admin Service (IISADMIN)</li> <li>HTTP SSL (HTTPFilter)</li> </ul>

hosts SQL Server Reporting Services	<ul> <li>SQL Server Reporting Services (TFSINSTANCE) (ReportServer)</li> </ul>
	<ul> <li>World Wide Web Publishing Service (W3SVC)</li> </ul>
Server that hosts SQL Server Analysis Services	• SQL Server Analysis Services
	<ul> <li>Internet Information Services Administration (IISADMIN)</li> </ul>
	• HTTP SSL (HTTPFilter)
Server that hosts SharePoint Products	<ul> <li>Windows SharePoint Services Timer (SPTimer) (Windows SharePoint Services 3.0 or Microsoft Office SharePoint Server 2007)</li> </ul>
	<ul> <li>World Wide Web Publishing Service (W3SVC)</li> </ul>
	<ul> <li>Microsoft Team Foundation Server</li> </ul>

	Application Pool
Application Application- pools tier server	<ul> <li>Microsoft Team Foundation Server Proxy Application Pool (only when Team Foundation Server Proxy is installed)</li> </ul>
	<ul> <li>DefaultAppPool (used by the Team Project portal)</li> </ul>
	⊠Note
Server that hosts SharePoint Products	The name might vary based on how SharePoint Products was installed.
	<ul> <li>SharePoint Central Administration v3</li> </ul>
	• Team Foundation Server
Web sites Application- tier server	• Team Foundation Server Proxy (only if Team Foundation Server Proxy is installed)
	• Default Web Site or Team Web site

	⊠Note
Server that hosts SharePoint Products	The name might vary based on how SharePoint Products was installed.
	• SharePoint Central Administration v3

**Required Permissions** 

To perform this procedure, you must be a member of the Administrators group on the server that hosts the service, application pool, or Web site that you want to manage. For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

## To stop or start a service, application pool, or Web site

- 1. Log on to the server that hosts the service, application pool, or Web site.
- 2. Click Start, click Administrative Tools, and then click Computer Management.
- 3. In the navigation pane, expand Services and Applications.
- 4. Perform one of the following steps based on what element you want to stop or start:
  - For a service, click Services, right-click the service, and then click Stop or Start.
  - For an application pool, click Internet Information Services (IIS) Manager, expand the local computer, and double-click Application Pools. Right-click the application pool, and then click Stop or Start.
  - For a Web site, click Internet Information Services (IIS) Manager, expand the local computer, and double-click Web Sites or Sites.

If the server is running Windows Server 2003, right-click the Web site, and then click Stop or Start.

If the server is running Windows Server 2008, right-click the Web site, point to Manage Web Site, and then click Stop or Start.

# **See Also**

#### Tasks

<u>Change the Service Account or Password for SQL Server Reporting Services</u> <u>Change the Service Account or Password for Team Foundation Server</u>

#### Concepts

Service Accounts and Dependencies in Team Foundation Server Team Foundation Server Concepts Team Foundation Server Architecture

#### **Other Resources**

How to: Verify or Correct the IIS Configuration for an Application Pool <u>Team Foundation Server Services and Service Accounts</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 Visual Studio Application Lifecycle Management
 Change Cache Settings for an Application-Tier Server
 See Also Send Feedback

You can help increase or balance performance in your deployment of Team Foundation Server (TFS) by changing the settings of the cache for files that are under version control on the application-tier server. By default, this cache is enabled so that users can download files more quickly from the cache instead of directly from the database. As an administrator, you can change the settings of this cache any time after installation.

You can change the following settings:

- Specify a different cache root folder.
- <u>Change the limit at which old files are removed from the cache.</u>

You can perform these tasks by editing the web.config file for version control, which is located in the installation directory on the application-tier server.

#### ⊠Note

By default, the installation directory for the application tier is %programfiles%\Microsoft Team Foundation Server 11.0\Application Tier\Web Services.

**Required Permissions** 

To perform these procedures, you must be a member of the Administrators security group on the application-tier server for Team Foundation.

For more information, see the <u>Microsoft website</u>.

## To specify a different cache root folder

1. On the application-tier server, create a cache folder.

You can create the folder on a local disk, in the UNC path, or on a mounted drive. For example, you might create the following folder:

d:\temp\cacheroot

#### Security Note

The cache folder stores sensitive information that is not encrypted. Therefore, you should make sure that only the service account of the application tier (TFSService) has Modify permissions to this folder.

2. Open the shortcut menu for the folder, and then choose Properties.

The Properties dialog box for the folder opens.

3. On the Security tab, choose Edit.

The Permissions dialog box opens.

4. Choose Add.

The Select Users, Computers, or Groups dialog box opens.

- 5. Add the local group **TFS\_APPTIER\_SERVICE\_WPG**, and then choose **OK**.
- 6. Select the **Modify** check box, clear all other check boxes, and then choose **OK**.
- In Windows Explorer (or File Explorer), browse to %programfiles%\Microsoft Team Foundation Server 11.0\Application Tier\Web Services.
- 8. Open the web.config file in a text or XML editor, and then locate the

<appSettings> section.

9. Add a line to the appSettings section to point to the folder that you just created:

## Copy Code

```
<add key="dataDirectory" value="NewCacheRootFolderLocation" />
```

For example, you would add the following line if you created a cache root folder that is named cacheroot in the temp directory of a hard drive, as in the earlier example:

#### Copy Code

```
<add key="dataDirectory" value="d:\temp\cacheroot" />
```

10. Save and close the web.config file.

#### ⊠Note

To maximize performance, copy the files from the old cache folder to the new cache folder.

- 11. Open a Command Prompt window, type **iisreset**, and then press ENTER.
- 12. Delete the old cache root folder.

#### ⊠Note

By default, the cache root folder is located at %programfiles%\Microsoft Team Foundation Server 11.0\Version Control Proxy\Web Services\VersionControlProxy\Data.

# ■Changing Limits for Removing Files from the Cache

You can change the maximum limit on the amount of storage space that the application-tier server can use for caching files. When this limit is reached, a cleanup routine makes room for newly requested files by deleting those files that have not been accessed in the longest time.

# To change the limit at which old files are removed from the cache

- 1. On the application-tier server, open Windows Explorer (or File Explorer), and browse to \%programfiles%\Microsoft Team Foundation Server 11.0\Application Tier\Web Services.
- 2. Open the web.config file in a text or XML editor, and then locate the <appSettings> element.
- 3. Add one of the following elements:
  - To specify a percentage of available disk space to fill before old files are removed, add the PercentageBasedPolicy element. You must specify a whole number as the value of this element.

For example, the following line specifies that the cache should fill up to 60% capacity of available disk space before old files are removed:

#### Copy Code

```
<add key="PercentageBasedPolicy" value="60" />
```

• To specify a fixed size in MB for the cache to reach before old files are removed, add the FixedSizeBasedPolicy element. You must specify a whole number as the value of this element.

For example, the following line specifies that the cache should reach

500 MB before old files are removed:

## Copy Code

<add key="FixedSizeBasedPolicy" value="500" />

⊠Note

If both the FixedSizeBasedPolicy and PercentageBasedPolicy elements are specified, the value of the FixedSizeBasedPolicy element is used, and the value of the PercentageBasedPolicy element is ignored.

- 4. Save and close the web.config file.
- 5. Open a Command Prompt window, type **iisreset**, and then press ENTER.

# To change the amount of cache to free when removing old files

- 1. On the application-tier server, open Windows Explorer (or File Explorer), and browse to %programfiles%\Microsoft Team Foundation Server 11.0\Application Tier\Web Services\.
- 2. Open the web.config file in a text or XML editor, locate the <appSettings> element, and then add the CacheDeletionPercent element.

For example, the following line specifies to free 50% of the cache when removing old files:

## Copy Code

```
<add key="CacheDeletionPercent" value="50" />
```

- 3. Save and close the web.config file.
- 4. Open a Command Prompt window, type **iisreset**, and then press ENTER.

# See Also

# Concepts

Service Accounts and Dependencies in Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 Visual Studio Application Lifecycle Management
 Move Team Foundation Server from One Hardware Configuration to Another Send Feedback

You can move TFS from one server to another by restoring it to the new hardware (also known as a restoration-based move). For example, you might want to move TFS to a server with greater capacity or improved processing speed, or you have concerns about the reliability of the hardware on which TFS is currently running. Moving TFS to a new server allows you to change the hardware that runs TFS without losing any of your project history.

To follow some of the steps in this procedure, you will need to have installed Update 2 or later for Team Foundation Server 2012. That update included the Scheduled Backups Wizard and the Restore Wizard, both of which are used as part of the move process. If for some reason you have not installed this update, you will need to follow the procedures for manually backing up and restoring your databases.

#### **▲Important**

In some situations you might want to change the domain of a TFS deployment as well as its hardware. Changing the domain is an environment-based move, and you should never combine the two move types. First complete the hardware move, and then change the environment.

To perform a restoration-based move, complete the procedures in the following sections:

- <u>Stop Services that Team Foundation Server Uses</u>
- Back up the Databases and the Encryption Key

- Install and Configure SQL Server on the New Data-Tier Server
- <u>Install and Configure Required Software on the New Application-Tier</u> <u>Server</u>
- <u>Restore the Databases</u>
- <u>Redirect SharePoint Products to the New Location of the Content Database</u>
- Install the Application Tier on the New Hardware
- <u>Update Team Foundation Server URLs</u>
- <u>Update All Service Accounts</u>
- <u>Update build servers</u>
- <u>Configure SharePoint Web Applications in TFS</u>
- <u>Configure Reporting and SQL Server Analysis Services</u>
- Configure backups
- <u>Verify Permissions for Users, Groups, and Service Accounts</u>
- <u>Configure Project Server Integration</u>
- <u>Refresh the Data Cache on Client Computers</u>
- <u>Update users</u>

#### **Required Permissions**

To perform the procedures in this topic, you must be a member of the following groups or have the following permissions:

- A member of the Administrators security group on the server or servers that are running the administration console for Team Foundation.
- Either a member of the SQL Server System Administrator security group or have your SQL Server Perform Back Up and Create Maintenance Plan

permissions set to Allow on the instance of SQL Server that will host the databases.

- A member of the sysadmin security group for the databases for Team Foundation and for the SQL Server Analysis Services database.
- An authorized user of the TFS\_Warehouse database.
- If the deployment uses SharePoint Products, a member of the Farm Administrators group for the farm to which you are restoring the databases for SharePoint Products.

# ■Stop Services that Team Foundation Server Uses

Stopping the services helps ensure that users cannot make changes to work items or check in source code to the original deployment during or after the move process.

## To stop services that Team Foundation Server uses

- 1. On the server that is running the application-tier services for Team Foundation, open a Command Prompt window, and change directories to Drive:\%programfiles%\Microsoft Team Foundation Server 11.0\Tools.
- 2. Type the following command:

TFSServiceControl quiesce

For more information, see

TFSServiceControl Command.

# Back up the Databases and the Encryption Key

Before you can move your deployment of Team Foundation Server, you must back up its databases. As part of the move, you will restore these databases to the new data-tier server. If you use reporting as part of your TFS deployment, the backup wizard will also give you the choice to back up the encryption key. Make sure that you do so, as you'll need it when restoring to the new server.

## ĭTip

The Scheduled Backups page referenced in this procedure is part of Update 2. If you have not installed the latest updates for TFS, consider doing so. If you cannot install the latest updates, or other operational considerations preclude you from using the wizard, then skip this procedure and take a full manual backup instead.

1. Open the administration console for TFS and on the Scheduled Backups page, take a full backup. The backup will back up everything you configured for backup in your backup plan, but it will do so immediately, not according to the time scheduled in the plan. If your deployment uses reporting, you can back up the encryption key as part of this backup set.



(If you don't have backups configured, you'll have to create a plan before you can take a full backup.)

2. Once the backup completes, verify that the backup is available on the storage device or network share, and that you can access this backup from the new hardware.

# Install and Configure SQL Server on the New Data-Tier Server

To restore data for Team Foundation, you must install SQL Server on the computer to which you want to move the databases for Team Foundation Server. The version of SQL Server that you install must exactly match the version on the original server that hosted the databases, including service pack level, collation settings, and language edition. If the match is not exact, you might not be able to restore the data or Team Foundation Server might not operate correctly even if you can restore the data.

If you want to use more than one server to host the databases for Team Foundation Server, you must install and configure SQL Server accordingly, either as a farm or as individual servers.

• Install SQL Server on the new server and make sure that it is operational. If your previous deployment used reporting, make sure that you include the reporting and analysis services components. You must install the same version and edition that you used previously, including service pack and cumulative update levels.

Features:	Description:	
Instance Features	Includes the Database Engine, the core service for storing, processing and securing data. The Database Engine provides controled access and rapid transaction processing and also provides rich support for sustaining high availability. The Database Engine also provides support for the utility control point in the SQL Server Utility. Only Database Engine Services and Analysis Services can be clustered.	

As an alternative, you can create an instance of SQL Server on a server that already has a matching version installed and restore the TFS databases to that instance, but that will require more post-restoration configuration.

For more information about options for installing and configuring SQL Server, go here.

After installing SQL Server, if your deployment includes reporting, open SQL Server Management Studio and detach the ReportServer and ReportServerTempDB databases. Otherwise you might not be able to restore these databases with the backup you created for the TFS databases.



# Install and Configure Required Software on the New Application-Tier Server

To configure a new server or servers to act as the new application-tier, you must first install and configure the software that is required to support Team Foundation Server. This software includes the following components:

- a supported operating system for your deployment configuration
- SharePoint Products (if your deployment is integrated with SharePoint Products and you want to move it to the same server as TFS)

#### ⊠Note

Unlike installing a new deployment of TFS, you will not be able to install SharePoint Products as part of the installation of the application tier when you move it to a new server. You must install the same version and edition of SharePoint Products that you used in your previous deployment.

• Install and configure Windows, IIS (if not configured by default), and SharePoint (if using) in the new environment, and make sure that the server and its software are operational.

For more information, see the system requirements for Team Foundation Server and Move SharePoint to New Hardware for TFS.

# **Restore the TFS Databases**

In order to restore the TFS databases using the restore tool, you must install but not configure TFS on the new data-tier server, and then use the restore function in the Scheduled Backups node.

If you want to restore TFS databases manually using SQL Server restoration tools, you can, but that is a more difficult procedure. In addition, you will have to manually unquiesce the databases in the new deployment. The restore wizard in TFS is part of Update 2. It automatically unqueisces the databases as part of its restoration process, but that functionality is not part of SQL Server's restoration tools.

If you installed SharePoint Foundation 2010 using the steps in Move SharePoint to New Hardware for TFS, and you plan on using that server as the server for TFS, the installation bits and administration console will already be present on the server and you can skip the first two steps in this next procedure.

- 1. Launch the TFS installation media. On the Team Foundation Server Setup page, choose Install.
- 2. When the installation completes, the Team Foundation Server Configuration Center opens. Close it.

The administration console opens automatically in an unconfigured state. This is expected.

3. To start the Restore wizard, open the administration console for TFS and open Scheduled Backups.



4. Specify the path to the backup set and choose the set you created after quiescing the old deployment.

Team Foundation Server Rest	ore Databases Wizard				
🌆 Restore D	atabases Wiza	ard			
Backup Sets Databases Review Readiness Checks Restore Databases	Select Backup Sets Point to the backup path and select the point in time you want to restore your T Server to. Note: The most recent backup has been selected by default.				
Complete	Network Backup Path:	\\fabrikambackup\TFS\	Browse		
	Date 3/14/2013 5:19:55 PM 3/14/2013 4:49:54 PM 3/14/2013 4:19:54 PM 3/14/2013 3:49:54 PM 3/14/2013 3:24:08 PM 3/14/2013 3:19:53 PM 3/14/2013 2:49:53 PM 3/14/2013 2:19:53 PM 3/14/2013 1:49:54 PM 3/14/2013 1:29:28 PM 3/14/2013 1:20:52 PM	Si 19 19 19 19 10 10 10 10 10 11 11 11 33 11 34 11	ize (MB) 92 92 91 91 91 90 00 00 80 79 79 8 8 78		

5. Complete the wizard and restore the databases to the new instance of SQL Server.


### ■Redirect SharePoint Products to the New Location of the Content Database

If your deployment uses SharePoint Products, you've already installed the same version and edition of SharePoint Products that you used in your previous deployment by following the steps in Move SharePoint to New Hardware for TFS, as mentioned above. Now, after you have restored the old deployment's content database (WSS\_Content) to the new server as part of the restoration set, you must redirect the server that is running SharePoint Products to the new location of that database. This database must be operational before you can reconfigure Team Foundation Server with the new locations of its databases.

- 1. Open a Command Prompt as an administrator on the new hardware that is running SharePoint Foundation.
- 2. Change directories to *Drive*:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\bin and run stsadm.exe with the following parameters, where SharePointFoundationServerName is the name of the server where you installed SharePoint Foundation 2010, and SQLServerName is the name of the server where you restored the WSS\_Content database as part of restoring TFS databases:

stsadm.exe -o addcontentdb -url http://SharePointFoundationServo

3. After that command completes successfully, type the following command, where Domain\UserName is the account you used to install and configure SharePoint Foundation 2010 for use with TFS:

stsadm.exe -o addpermissionpolicy -url http://SharePointFoundat:

## Install the Application-Tier Server

- 1. From the administration console for TFS, choose Configure Installed Features to launch the configuration center.
- 2. Launch the Application-Tier Only wizard, and in Databases, specify the new SQL Server instance where you restored the TFS databases. Choose the Tfs\_Configuration database from the list.

	Team Foundation Ser	ver Application Tier Only Wizard	
🙀 Application Tier Only Wizard			
Welcome Databases Application Tier Settings Review Readiness Checks Configure Complete	Specify a Team I Type in a SQL Server in SQL Server Instance: NEWFABRIKAMTFS Databases: Database Name Tfs_Configuration	Foundation Server configuration datab nstance path, click List Available Databases, then sele Version Microsoft Team Foundation Server 2013 RC Con	

3. Before you close the final page of the wizard, look for the "i" symbol. It signifies information that you might want for future reference. The final page also includes the location of the configuration log.

# ■Update Team Foundation Server URLs

1. Go to the application-tier node and look at the notification and Team Web Access URLs. Note that they still point to the location of the old deployment. Update them.

ዀ Applicati	on Tier	🔹 Refresh	(?) Help
Application Tier Su	nmary		
Service Account: Web Site: Application Pool:	NT AUTHORITY\NETWORK SERVICE Team Foundation Server Microsoft Team Foundation Server Application Pool	Generation Change	Password Account Account
Notification URL: Server URL: Web Access URL: Machine Name:	http://fabrikam2008:8080/tfs http://localhost:8080/tfs http://fabrikam2008:8080/tfs NEWFABRIKAMTFS	Change	Membership ster Security URLs
Ports: Virtual Directory: Version:	8080 /tfs 12.0.20812.0 (Tfs2013.RC)		

2. After updating the URLs with the name of the new server, review the information to make sure that it is correct.

Iteam Foundation Server Administration Console			
File Help			
<ul> <li>NEWFABRIKAMTFS</li> <li>Application Tier</li> <li>Team Project Collections</li> <li>SharePoint Web Applications</li> <li>Reporting</li> <li>Lab Management</li> <li>Proxy Server</li> <li>Build Configuration</li> <li>Extensions for SharePoint Products</li> <li>Scheduled Backups</li> <li>Logs</li> <li>Additional Tools and Components</li> <li>Visual SourceSafe Upgrade</li> <li>PreEmptive Analytics</li> </ul>	Application Tier Sum     Application Tier Sum     Service Account:     Web Site:     Application Pool:     Authentication:     Authentication URL:     Server URL:     Web Access URL:     Machine Name:     Ports:     Virtual Directory:	Tier Mmary NT AUTHORITY\NETWORK SER Team Foundation Server Microsoft Team Foundation Se Pool NTLM http://newfabrikamtfs:8080/tfs http://newfabrikamtfs:8080/tfs http://newfabrikamtfs:8080/tfs NEWFABRIKAMTFS 8080 /tfs	

### **Update All Service Accounts**

You must update the service account for Team Foundation Server (TFSService) and the data sources account (TFSReports). Even if these accounts have not changed, you must update the information to help ensure that the identity and the format of the accounts are appropriate for the new server.

- 1. Open a Command Prompt window as an administrator and change directories to Drive:\%programfiles%\Microsoft Team Foundation Server 11.0\Tools.
- 2. At the command prompt, type the following command to add the service account for Team Foundation, where DatabaseName is the name of the configuration database (by default, TFS\_Configuration):

**TfsConfig Accounts /add /AccountType:ApplicationTier** /account:AccountName /**SQLInstance**:ServerName /**DatabaseName**:DatabaseName

3. At the command prompt, type the following command to add the data sources account:

TfsConfig Accounts /add /AccountType:ReportingDataSource /account:AccountName /SQLInstance:ServerName /DatabaseName:DatabaseName

For more information, see

Accounts Command.

## **Update build servers**

Now you'll need to redirect your build servers to point to the moved TFS deployment.

- 1. On each build server, open the administration console and stop the build service.
- 2. In the properties for the build service, update the communications properties.

Build Configuration	🙇 Refresh	🕜 Hel
Build Service configured for http://fabrikam2008:8080/tfs/def \NETWORK SERVICE.	aultcollection as NT AUTH	ORITY
FABRIKAMTFS - Stopped on http://fabrikamtfs.redmond.corp.microsoft.com AUTHORITY\NETWORK SERVICE Start   Properties   Unregister	9191/Build/v5.0/Services a	is NT
Build Service Properties	? 🔀	
Stop the service to make changes Communications: Provide Build Services for Project Collection:		
http://fabrikam2008:8080/tfs/DefaultCollection	Browse	
Listen for Build Agent communication on:		
http://fabrikamtfs.redmond.corp.microsoft.com:9191/	Change	
Run the Service as:		
NT AUTHORITY\NETWORK SERVICE	Change	
Connect to Team Foundation Server as:		
Use the same identity as Windows Service		
	Change	
Due the Service Interactively when is this weeful?		

# ■Configure SharePoint Web Applications

If your deployment uses SharePoint Products and you moved SharePoint Foundation 2010 as part of the TFS move, you might need to redirect TFS to the new web application. Even if you don't, you should still repair the connection to help ensure proper performance.

If you're not using SharePoint Products as part of your deployment, or if your deployment will continue to use the old SharePoint server, you can skip this procedure.

• Open the administration console and navigate to SharePoint Web Applications. If the web application still refers to the old site, or if your new deployment uses a different web application than the one listed, choose Change and update the settings.

	a SharePoint Web Applicat	ions	💰 Refresh	
^	<ul> <li>SharePoint Web Applications</li> <li>You can use SharePoint Web applications to provide collaboration resources for members of a team project, such as team project portals and document libraries. Before you can specify a SharePoint Web application, you must install the Team Foundation Server Extensions for SharePoint Products and configure them for at least one Web application. More Information</li> <li>Name</li> </ul>		<ul> <li>♣ Add</li> <li>✓ Change</li> <li>★ Remove</li> <li>♣ Repair Connect</li> </ul>	
^	http://fabrikam2008   Service Accounts for SharePoint Web A	http://fabrikam2008		
	You must specify the service accounts for support team project portal functionality service account for both the Web applicat site are required.	SharePoint Web applications to for Team Foundation Server. The ion and Central Administration	\mu Add Members 🗙 Remove	

If the information is correct, or after you have corrected it, choose Repair Connection. This helps ensure that everything is working properly.

## ■Configure Reporting and Analysis Services

If your deployment uses a report server, you must redirect Team Foundation Server to its location, restart the warehouse, and manually rebuild the database for Analysis Services. If you don't use reporting, skip this procedure.

- 1. Go to the Reporting node. The listed report server values are the old ones, not the new, so edit them.
- 2. Change the values on all three tabs to point to the new server. Make sure that you provide the correct information for the data sources account in the new deployment.
- 3. Choose Start Jobs to restart reporting.
- 4. Choose Start Rebuild to rebuild the warehouse.

# **Configure backups**

Although you had backups scheduled for your old deployment, those scheduled backups weren't changed to back up your moved deployment. You'll need to configure them.

• In the administration console, go to the Scheduled Backups node and reconfigure the scheduled backups to back up the TFS databases on the new server. For more information, see

Create Back Up Schedule and Plan.

## ■Verify Permissions for Users, Groups, and Service Accounts

After you move to new hardware, make sure that all users, groups, and service accounts for your deployment are configured with the permissions that they require to function correctly on each server. Some permissions, such as additional permissions in SQL Server or on the local computer, cannot be automatically migrated. For example, Team Foundation administrators must be members of the local Administrators group on the application-tier server to open the administration console, so you must add manually them to that group.

- Log on to the server and make sure that users, groups, and service accounts are configured with the permissions required for operation. Manually spotcheck membership in project groups and teams, and verify that those groups and teams have the permissions you expect.
- Browse to a team project collection and make sure that all projects in that collection appear as expected, and that users in those projects can appropriately access their work items.
- Open Team Web Access and verify that team sites and teams appear as expected.

Not sure what groups and permissions to expect? For more information, see

<u>Add Users to Team Projects, Set Administrator Permissions for Team Project</u> <u>Collections, Set Administrator Permissions for Team Foundation Server, and</u> <u>Service Accounts and Dependencies in Team Foundation Server</u>.

### **Configure Project Server Integration**

You can skip this procedure if the deployment that you are moving is not integrated with Microsoft Project Server. For more information, see Enable Data Flow Between Team Foundation Server and Microsoft Project Server.

If you intend to continue integration between the projects and collections and Project Server, you must register the instance of Project Web Access or Project Web App (PWA) with the application-tier server for your moved deployment.

### To integrate the moved deployment with Project Server

 Register an instance of Project Web Access or Project Web App with the moved deployment of Team Foundation Server by using the **TFSAdmin ProjectServer /RegisterPWA** command with the /tfs, /force, and /pwa options.

After you have registered the instance, you must wait for the data to synchronize. This process happens automatically on a predetermined schedule. For more information, see Overview of the Synchronization Process for Team Foundation Server and Project Server Integration, Change the Configuration of Your Deployment, Register an Instance of PWA to Team Foundation Server, and Remove a Component from Participating in Data Synchronization.

# ■Refresh the Data Cache on Client Computers

### To refresh the data cache on client computers

• Log on to the server, and use the ClientService Web service to force clients to update the cache for tracking work items and for version control.

For more information, see

Refresh the Data Caches on Client Computers. **Note** 

If you restored your databases to a different point in time, you will also need to refresh the version control cache.

## **Update users**

Now that you've moved TFS, you'll need to tell your users how to connect to the moved deployment. Specifically, you'll need to give them the following information:

- The name of the new server and the URL for Team Web Access, so that they can reconnect to their projects
- The new database names for reporting, if reporting is part of your deployment
- The new URL for SharePoint, if SharePoint is part of your deployment

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Move Team Foundation Server from One Environment to Another
See Also Send Feedback

The most common environment-based move scenario is changing the domain of the TFS deployment, whether it's a domain name change or going from a workgroup to a domain.

### **▲Important**

In some situations you might want to change the domain of a TFS deployment as well as its hardware. Changing the hardware is a restoration-based move, and you should never combine the two move types. First complete the hardware move, and then change the environment.

Additionally, changing identities in TFS as part of an environmental move is the aspect that most often causes conflicts or problems. The <u>Identities Command</u> is a powerful tool, but it has certain limitations. Read about it as part of planning your move. To help ensure a successful move, make sure that you understand the following requirements:

- Once a user account is present in TFS, it cannot be removed or have another account mapped to it. For example, if you are moving DomainA/UserA to DomainB/UserB, the Identities command would only work to migrate the user if DomainB/UserB is not already present in TFS.
- Because the members of the local Administrators group are automatically added to TFS, make sure to remove any accounts that you want migrated from that group before you change the domain or environment.

For further background information, <u>go here</u>.the There's a detailed description of how identity changes in TFS work, including limitations of the tool.

We'll walk through the steps to change the environment of your TFS deployment in the following sections:

- 1. Check permissions and accounts
- 2. <u>Stop TFS services</u>
- 3. Back Up Data
- 4. Join TFS to its new domain
- 5. Configure SharePoint Products for the new environment
- 6. Move TFS user and service accounts
- 7. Configure Reporting and Analysis Services
- 8. <u>Restart TFS services</u>

# **Check permissions and accounts**

To successfully change the environment for TFS, you'll need to be an administrator on the local computer as well as for TFS and all of the software on which your deployment depends: SQL Server, reporting, SharePoint Products (if your deployment uses reporting or SharePoint), and any other software with which your deployment interoperates, such as Project Server. However, all members of the local Administrators group are automatically included in TFS, which can cause problems when trying to migrate accounts. Therefore, you should use an account that you do not intend to migrate as part of the environmental move. You might consider adding a special administrative account just for the move, and using that account to perform the migration.

### To verify administrator-level permissions

- Make sure the account you're using is a member of the following groups:
  - Servers: Administrators (local Administrators group or equivalent)
  - TFS: Team Foundation Administrators and Admin Console Users
  - SQL Server: sysadmin
  - SharePoint Products: Farm Administrators (if your TFS deployment integrates with SharePoint Products)

If you aren't a member of one or more of these groups, get permissions now.

Now that you're sure you're using an account that has all the permissions needed, it's time to start checking accounts to see if there might be any conflicts with names or groups in the environment to which you'll be moving. We already know that accounts that are members of the local Administrators group can't be migrated, so let's remove those first.

### Remove accounts to be migrated from local Administrators group

• Open the local Administrators group and remove any accounts that you

wish to migrate to the new environment. Repeat this step for any other groups that might be affected.

Now check the list of identities in the current TFS environment and look for any potential problems with groups or individual user accounts that might exist in the new environment.

### ĭTip

Consider creating a table or migration map of identities to be moved as part of the environmental move, including details of which accounts might not be able to be migrated automatically.

### **Check identities**

 On the application-tier server for Team Foundation, open a Command Prompt window with administrative permissions, navigate to %ProgramFiles%\Microsoft Visual Studio 11.0 Team Foundation Server\Tools, and run this command to view the identities currently in the system:

#### Copy Code

TFSConfig Identities

2. A list of identities will appear. Check these users and groups to ensure that there are no potential duplicates or problems with identities in the environment to which you'll move TFS, and take steps to mitigate any potential conflicts.

### **Stop TFS services**

Stopping the services helps ensure that users cannot make changes to work items or check in source code to the original deployment during or after the move process.

- 1. On the TFS application-tier computer, open a Command Prompt window, and change directories to Drive:\%programfiles%\Microsoft Team Foundation Server 11.0\Tools.
- 2. Type thisTFSServiceControl command:

TFSServiceControl quiesce

### Back Up Data

### To back up data for Team Foundation

• For more information, see

<u>Manually Back Up Team Foundation Server</u> and How to: Back Up the Reporting Services Encryption Key.

### **Join TFS to its new domain**

- 1. On each server, open the properties for the computer.
- 2. Change the settings for the computer to the domain or workgroup to which you want to join the server.

If you are prompted to provide the user name and password of an account that has permissions to join this computer to the domain, provide the appropriate credentials.

3. Restart the computer for the domain change to take effect.

#### ⊠Note

After you restart the computer, a warning might appear that services or drivers could not be started. Continue with the next procedure.

## **Configure SharePoint Products for** the New Environment

If you are changing the environment to one where there is no trust with your previous environment, you might need to configure SharePoint Products before it will operate correctly in the new environment. Information about users imported from directory services is available on SharePoint sites from the Select People and Groups dialog box, also known as the People Picker Web control. Site administrators and other users use the People Picker to select people and groups when assigning permissions. When information about users is located on multiple forests or on a forest without a trust relationship for all users, additional steps might be necessary to ensure that all people and groups are available from this Web control.

Skip this procedure if you are not using SharePoint Products in your deployment, if your new environment has a two-way trust to the old environment, or if no errors for your SharePoint Web application appear in the administration console for Team Foundation.

1. On every server that is part of the SharePoint farm that supports your deployment of Team Foundation Server, open a Command Prompt window, and change directories to %COMMONPROGRAMFILES%\microsoft shared\web server extensions\14\bin.

#### Caution

The number in the path will vary, depending on the version of SharePoint Products you are using.

2. Type this command, where Key is the encryption key you want to use in your deployment of SharePoint Products:

#### stsadm.exe -o setapppassword -password Key

ĭNote

This key is an encryption string that is used to encrypt the password for the account that is used to access the forest or domain. The encryption string must be the same for every server in the farm, but a unique string should be used for each farm.

3. Type this command, where domain:DNSName is the target forest or domain and its DNS name, user,password is the username and password for an account that has access to the target forest or domain, and WebApp is the name of the Web application that supports your deployment of Team Foundation Server:

### **stsadm.exe -o setproperty -pn peoplepicker-searchadforests -pv** domain:DnsName,user,password **-url http:**//WebApp

4. Type this command, where URL is the URL for a site collection that supports a team project collection, Port is the port number that is assigned to that site collection, and UserName is the user name of the account that will act as the owner for that site collection:

### stsadm.exe -o siteowner -url http://URL:Port -ownerlogin UserName

5. Repeat the previous step for each site collection that your deployment of Team Foundation Server uses.

### **Move TFS user and service accounts**

As mentioned at the beginning of this topic, moving accounts is when you're most likely to encounter difficulties, particularly if you haven't carefully planned for user migration. The TFSConfig Identities command cannot migrate any account to an account that already exists in TFS.

If account names are the same in both domains, and the only difference is the domain name, then you can use the batch mode of TFSConfig Identities to change all the identities at once. Otherwise you must change identities individually and specify a different target account name, as detailed next.

 On the application-tier server for Team Foundation, open a Command Prompt window with administrative permissions, navigate to %ProgramFiles%\Microsoft Visual Studio 11.0 Team Foundation Server\Tools, and run this command to change the service IDs (SIDs) for the service account to the new domain:

TFSConfig identities /change /fromdomain:OldComputerorDomainNam

#### Caution

If your service account was a system account such as Network Service, you cannot directly migrate the service account, because a system account with the same name exists in the new environment. You'll have to perform a two-stage process change. See the example in

#### Identities Command.

• To migrate all accounts that have the same name in the new environment, type this command:

Copy Code

TFSConfig Identities /change /fromdomain:OldDomainName /todomain:Nev

This will batch process the accounts.

• If your new domain contains one or more identities where the name changes between environments, you'll need to manually update the SIDs for each of those identities. For example, if Christie Church's user account was Fabrikam\CChurch in the previous environment, but is NewFabrikam\ChristieC in the new environment, you would have to manually update her SID. For every account that has this requirement, type this command:

#### 

TFSConfig Identities /change /fromdomain:OldDomainName /todomain:Nev

• Now run this command to update the service account:

#### Copy Code

TFSConfig Accounts /change /AccountType:ApplicationTier /account:Acc

• If your deployment uses reporting, run this command to update the data source account used for reporting:

#### Copy Code

TFSConfig Accounts /change /AccountType:ReportingDataSource /account

• If your deployment uses Team Foundation Server Proxy, run this command to update the service account used for the proxy:

#### Copy Code

TFSConfig Accounts /change /AccountType:Proxy /account:AccountName /

ĭNote

If you are moving to a non-trusted domain, you might also need to manually add users and groups to teams, projects, collections, and Team Foundation Server itself. For more information, see <u>Add Users to Team Projects</u>, <u>Set Administrator Permissions for Team Project Collections</u>, and <u>Set Administrator Permissions for Team Foundation Server</u>.

• If your deployment is integrated with Project Server, you might need to perform additional steps to configure the service accounts with the permissions required for operation. For more information, see Assign Permissions to Support Integration of Project Server and Team Foundation Server and Configure the Integration of Team Foundation Server and Project Server.

## ■Configure Reporting and Analysis Services

You can skip this procedure if you are not using reporting as part of your deployment. If you renamed a report server as part of this type of move, you must redirect Team Foundation Server to the report server at its new location. You must also restart the warehouse and manually rebuild the database for Analysis Services.

#### ⊠Note

If an error appears stating that that the account cannot be added to the TFSEXECROLE or TfsWarehouseDataReader role, the account is not valid in the new environment. You must remove the old account by using the TFSSecurity command-line tool with the /g- command, and update it with an account that is valid in the new environment. For more information, see

Changing Groups and Permissions with TFSSecurity and /g- Command.

### To reconfigure reporting and Analysis Services

- 1. Open the administration console for Team Foundation.
- 2. In the navigation bar, click Reporting.
- 3. On the Reporting page, click Edit.
- 4. In the Take Offline dialog box, click OK.

The Reporting Services dialog box opens.

- 5. Select the Use Report Server check box.
- 6. Click the Warehouse tab, and, in Server, type or click the new name of the server.

7. In Database, type the name of the warehouse database for Team Foundation Server.

By default, this database is named TFS\_Warehouse.

- 8. (Optional) Click Test Connection to make sure that the database that you specified is valid.
- 9. Click the Analysis Services tab.
- 10. In the Server list, type or click the new name of the server that hosts the Analysis Services database.
- 11. In Database, type the name of the Analysis Services database for Team Foundation Server.

By default, the name of this database is TFS\_Analysis.

- 12. If you are not using the default instance for the database, select the Specify nondefault instance check box, and then type or click the name of the instance.
- 13. (Optional) Click Test Connection to make sure that the database that you specified is valid.
- 14. In Username and Password, type the account name and password (if any) for the data sources account (TFSReports).
- 15. On the Reports tab, in the Server list, type or click the name of the report server, and then click Populate URLs.
- 16. In Username and Password, type the account name and password (if any) for the data sources account (TFSReports).
- 17. In Default Path, type the relative path for storing reports, and then click OK.
- 18. In the administration console, click Start Jobs to restart reporting.
- 19. Open a Command Prompt window, and change directories to

%ProgramFiles%\Microsoft Team Foundation Server 2012\Tools.

20. Type this command to rebuild the database for Analysis Services:

TFSConfig RebuildWarehouse /AnalysisServices /ReportingDataSourcePassword:Password

Password is the password for the data sources account for Reporting Services (TFSReports).

- 21. Wait until the command is successfully completed.
- 22. On the report server, open Internet Explorer, type this string in the Address bar, and then press ENTER:

#### http://localhost:8080/tfs/TeamFoundation/Administration/v3.0/Warehou

The ControllerService page opens.

23. Click GetWarehouseStatus, and then click Invoke.

#### **▲Important**

The service should return a value of Idle, which indicates that the cube is not being processed. If a different value is returned, repeat this step until Idle is returned.

24. Return to the ControllerService page, click Run, and then click Invoke.

This step initiates processing. The service returns True when it has successfully started the controller and False if the warehouse is still being processed.

25. To determine when the warehouse has been processed, return to the ControllerService page, click GetWarehouseStatus, and then click Invoke.

Processing is completed when the GetWarehouseStatus service returns a value of Idle.

### ■Restart Services That Team Foundation Server Uses

### To restart services that Team Foundation Server uses

- 1. On the server that is running the application-tier services for Team Foundation, open a Command Prompt window, and change directories to Drive:\%programfiles%\Microsoft Team Foundation Server 2012\Tools.
- 2. Type this command:

TFSServiceControl unquiesce

For more information, see

TFSServiceControl Command.

## **Configure Project Server Integration**

You can skip this procedure if the deployment that you are moving is not integrated with Microsoft Project Server.

- 1. Register an instance of PWA with the moved deployment of Team Foundation Server by using the **TFSAdmin ProjectServer /RegisterPWA** command with the **/tfs**, **/force**, and **/pwa** options.
- 2. Wait for the data to synchronize.

This process happens automatically on a predetermined schedule. For more information, you can read more about TFS integration with Project Server

<u>here</u>..

### See Also

#### Tasks

Move Team Foundation Server from One Hardware Configuration to Another

### Concepts

Team Foundation Server Architecture Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Configuring Users, Groups, and Permissions
Send Feedback

Team Foundation security is based on users and groups. You can use the default groups in Visual Studio Team Foundation Server (TFS) to manage users and groups as part of implementing a security model for your organization. Default groups exist at the server level, at the collection level, and at the project level. You can also create custom groups at any of these levels with specific permissions to better fit your security model. Using a group-based security model enables users to access the data that they require without requiring the granting of specific permissions to each user account. This strategy helps protect confidential information while reducing the administrative overhead of managing user access to the deployment.

For each role in your business, you must determine what group memberships users require to accomplish their tasks. TFS, SharePoint Products, and SQL Server Reporting Services all maintain their own information about groups, users, and permissions. You must carefully plan how you want to manage users and permissions. This planning applies not only across individual projects in TFS but also across TFS itself, the Windows operating system, and if configured for your deployment, SharePoint Products and SQL Server Reporting Services. Within TFS, you can allow or deny permissions at many levels, such as for work items, access to version control, access to a project or project collection, and access to TFS itself.

In specific cases, you might want to add a user directly to a team project or team project collection, instead of adding the user to a group at the project level or collection level. In that case, you must grant permissions directly to the user account for that user.

### **Common Tasks**

	Title	Description
		<u>Add Users to</u> <u>Team</u> <u>Projects</u>
	Add users or groups of users: You can quickly add users to team projects, team project collections, or TFS itself by adding them to the	Set Administrator Permissions for Team Project Collections Set Administrator Permissions for Team
	appropriate default groups for their roles.	<u>Foundation</u> <u>Server</u> Configure
		<u>Team</u> <u>Foundation</u> <u>Server to</u> <u>Support Your</u> <u>Development</u> <u>Teams</u>
	Change or modify default groups: You can change the membership	<u>Team</u> <u>Foundation</u> <u>Server</u> <u>Permissions</u>

of default groups. You can also modify the default permissions that Server are granted to some of these groups to better meet your security needs.

Team **Foundation** Default Groups, Permissions, and Roles

Team Foundation <u>Server</u> Permissions

#### Team

Understand and manage permissions: You can review all individually configurable permissions within Team Permissions, Foundation Server, learn and Roles what permissions are assigned by default, and view the permissions for specific groups or users.

**Foundation** Server Default Groups,

**Change** access levels

<u>View</u> **Permissions** 

Change **Permissions** for a Group or User

Service Accounts and **Dependencies** in Team Foundation
Understand service accounts and groups: In addition to managing permissions for your users, you must also manage the permissions that are required by the service accounts upon which Team Foundation Server Server depends.

<u>Server</u>

Change the **Service** Account or Password for <u>Team</u> **Foundation** 

Change the **Service** Account or Password for SQL Server **Reporting Services** 

# **Related Sections**

Administering Team Foundation Server

Get Up and Running With a Single-Server Installation [Tutorial]

Collaborate

Collaborate (dig deeper)

Planning and Tracking Projects

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Add Users to Team Projects
See Also Send Feedback

You can control the level of access that various members of your team have to team projects by assigning each member to groups for each project. To provide team members with full access to all the functionality of your project, you might also have to add them to groups in SQL Server Reporting Services and SharePoint Products.

#### **▲Important**

Even after you configure your users with the permissions that they need, your users might not have access to certain features in Team Foundation Server. To enable those features, you might need to <u>Change access levels</u>.

By default, Visual Studio Team Foundation Server creates several groups for each project, and each group has its own level of permissions and rights for that project. You can use these default groups or create custom groups that have specific permissions. For example, you can add users to the Readers group so that they can view work items for a project but not create, modify, or delete work items. You can also allow other users to update work items but not to perform administrative tasks, such as deleting projects and publishing test results. For more information about the default user groups in Team Foundation Server, see <u>Team Foundation Server Permissions</u>.

If your project uses resources on SharePoint Products and Reporting Services, you must add users who require those resources to groups that have appropriate permissions to those programs. To grant users full access to all functionality for your project, you must add them to the default groups in the following table:

Software	Readers	Contributors	<b>Project Leads</b>
Team Foundation Server	Readers	Contributors	Project Administrators
SharePoint Products	Visitors	Members	Owners
SQL Server Reporting Services	Browser	Browser	Team Foundation Content Manager

#### ĭTip

Unlike Team Foundation Server and SharePoint Foundation 2010, SQL Server Reporting Services does not distinguish between projects. Therefore, if you add a group to Reporting Services, that group will have the same permissions for reports across all the projects in the collection, regardless of their permissions in individual projects. Keep this in mind when choosing what groups to add.

For information about how to set permissions for administrators in Team Foundation, see <u>Set Administrator Permissions for Team Project Collections</u> and <u>Set Administrator Permissions for Team Foundation Server</u>.

#### **Required** permissions

To perform these procedures, you must be a member of the following groups:

- the Project Administrators group or your Edit Server-Level Information permission must be set to Allow in Team Foundation Server
- the Site Administrators group or Site Collection Administrators group in SharePoint Products

- Content Managers group or the Project Content Managers group in SQL Server Reporting Services
- To verify or change a user's membership in a licensing group, you must be a member of the Team Foundation Server Administrators group.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

# To add users or groups in Team Foundation Server

1. In Team Explorer, open the team project to which you want to add users, and then choose Settings.

The Settings menu appears.

2. In the Settings menu, in Team Project, choose Group Membership.

Team Web Access opens in the administration context for the team project, on the security tab.

- 3. On the groups tab, choose one of the following:
  - To add users who will require minimal access to the project, choose Readers.
  - To add users who will contribute fully to this project, choose Contributors.
  - To add users who will act as project leads, choose Project Administrators.

Once you have chosen one of those groups, on the members tab, choose Add, and then choose Add user.

The ADD A WINDOWS USER OR GROUP window opens.

4. In Identities, specify the name of the user or group you want to add.

ĭTip

The first time you add a user or group to Team Foundation Server, you cannot browse to it or check the name. After the identity has been added anywhere in Team Foundation Server, you will be able to use those functions to quickly find the name.

- 5. When you have finished adding the groups or users you want to add, choose Save Changes.
- 6. Depending on the user, you might want to customize their permissions for other functionality in the project, such as <u>areas and iterations</u> or <u>shared</u> <u>queries</u>. You can also control access to projects, version control, build, and work items; learn how in Restrict access in TFS.

## To add users or groups in SharePoint Foundation 2010

1. In Team Explorer, on the Team menu, choose Show Project Portal.

The portal for the team project opens in a separate window.

#### ĭTip

If your deployment uses a different version of SharePoint Products, consult the <u>guidance for your version</u> to get the current steps for managing users in SharePoint.

2. Choose Site Actions, and then choose Site Permissions.

The browser window changes to Permissions Tools.

3. Choose Grant Permissions.

The Grant Permissions window opens.

- 4. In Users/Groups, specify the name of the group that you want to add. In Grant Permissions, choose Grant users permission directly, and then select one of the following check boxes:
  - To add users who will require minimal access to the project, select the

Readers check box.

- To add users who will contribute fully to this project, select the Contributors check box.
- To add users who will act as project leads, select the Full Control check box.
- 5. When you have finished adding the groups or users you want to add, choose OK.

## To add users or groups in SQL Server Reporting Services

- 1. Open Internet Explorer or another browser compatible with Reporting Services administration.
- 2. Type the following into the address bar, where ReportServer represents the name of the server that is running SQL Server Reporting Services:

#### http://ReportServer/Reports/Pages/Folder.aspx

- 3. On the Home page, choose Folder Settings.
- 4. On the Security page, choose New Role Assignment.
- 5. On the New Role Assignment page, in Group or use name, specify the name of the group you want to add, and then select one of the following check boxes:
  - To add users who can act as readers of or contributors to the project, select the Browsers check box.
  - To add users who will act as project leads, select the Team Foundation Content Manager check box.

Once you have finished adding the user or group and selected the appropriate check box, choose OK.

### To verify features available for a user

- 1. Open Team Web Access in administration mode, and choose Control Panel to navigate to the top-level administration context.
- 2. Choose the web access tab.
- 3. In Web Access Permissions, choose the name of each license group in turn, and review the following information:
  - 1. Which licensing group is set as the default group for the deployment. That group name will be followed by (Default). This is the group that all users of your deployment of Team Foundation Server will be assigned to by default.

#### ĭTip

The default group is set to Standard.

- 4. Whether the user for whom you are determining licensing levels is a member of a different licensing group than the default group. If so, review the description of the features of that licensing group to better understand what features are and are not available to that user.
- 5. To review the licensing group membership of all users in the deployment at once, choose Export Audit Log. The membership information will be exported to a comma-delimited file. Save or open the file.

For more information about licensing levels and the features available in Team Foundation Server, see the Team Foundation Server Licensing White Paper.

# **See Also**

#### Tasks

<u>Set Administrator Permissions for Team Foundation Server</u> <u>Set Administrator Permissions for Team Project Collections</u>

#### Concepts

<u>Team Foundation Server Permissions</u> <u>Team Foundation Server Default Groups, Permissions, and Roles</u> <u>Configuring Users, Groups, and Permissions</u>

#### **Other Resources**

Server-Level Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Set Administrator Permissions for Team Project Collections
See Also Send Feedback

To grant administrator permissions to users for a team project collection, you add them to appropriate groups or grant them specific permissions in Visual Studio Team Foundation Server. If your collection uses resources on SharePoint Products and SQL Server Reporting Services, you also grant those permissions to users in those programs.

Project collection administrators maintain the collection and administer permissions and security for other roles at the collection level. Members of the Project Collection Administrators group have the highest set of permissions of any users in the collection and the second-highest set of permissions of any default group in Team Foundation Server. For most organizations that use Team Foundation Server, project collection administrators manage the collections that members of the Team Foundation Administrators group create, but members of the Project Collection Administrators group do not create the collections themselves. Project collection administrators also perform any operations that are required to maintain the collection, such as creating team projects, adding users to groups, or modifying the settings for the collection.

For information about how to set permissions for other members of the team, see <u>Add Users to Team Projects</u>.

An administrator for team project collections must belong to the following groups:

• Team Foundation Server: Project Collection Administrators or the appropriate collection-level permissions set to Allow.

For more information, see <u>Team Foundation Server Default Groups</u>, <u>Permissions, and Roles</u>.

• SharePoint Products: If the collection is configured with a site collection resource, see Site Collection Administrators

This group might vary depending on your version of SharePoint Products and the security requirements of your deployment. For more information, see Interactions Between SharePoint Products and Team Foundation Server and Roles in SharePoint Products.

• Reporting Services: If the collection is configured with reporting resources, see Team Foundation Content Manager

#### ⊠Note

Even users who have appropriate permissions might not be able to view team project portals or reports correctly until they add the sites to their Trusted Sites in Internet Explorer. For more information, see this topic on the Microsoft Web site: <u>User Account Control</u>.

#### **Required Permissions**

To perform these procedures, you must either have the following permissions or belong to the following groups:

• Project Collection Administrators group or the View Server-Level Information and Edit Server-Level Information permissions must be set to Allow

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- To add permissions for SharePoint Products, the Site Collection Administrators group or the Farm Administrators group for SharePoint Products
- To add permissions for Reporting Services, the Content Managers group or the Project Content Managers group for Reporting Services

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

In addition to these permissions, you might need to address the following requirements on a computer that is running Windows Server 2008 or Windows Vista:

- To follow a procedure that requires Internet Explorer, you might need to start it as an administrator by clicking Start, clicking All Programs, right-clicking Internet Explorer, and then clicking Run as administrator.
- To access Report Manager, reports, or Web sites for SQL Server Reporting Services, you might need to add these sites to the list of trusted sites in Internet Explorer or start Internet Explorer as an administrator.

For more information, see this topic on the Microsoft Web site: <u>User Account</u> <u>Control</u>.

# To add users or groups in Team Foundation Server

1. In Team Explorer, choose the Team menu, choose Team Project Collection Settings, and then choose Security.

Team Web Access opens in the administration context for the team project, on the security tab.

2. On the groups tab, choose Project Collection Administrators.

Choose the members tab, choose Add, and then choose Add Windows user or group.

The ADD A WINDOWS USER OR GROUP window opens.

3. In Identities, specify the name of the user or group you want to add.

ĭTip

The first time you add a user or group to Team Foundation Server, you cannot browse to it or check the name. After the identity has been added anywhere in Team Foundation Server, you will be able to use those functions to find the name.

4. When you have finished adding the groups or users you want to add, choose Save Changes.

# To grant administrative permissions in SharePoint Foundation 2010

1. On the server that is running SharePoint Products, open SharePoint Central Administration.

For more information, see Access Site Administration or Central Administration for SharePoint Products.

2. Grant permissions that are appropriate for this user at the farm or the Web application level, depending on your security needs.

For more information, see Windows SharePoint Services Roles and the following page on the Microsoft Web site: <u>SharePoint Products Tech</u> <u>Center</u>. For optimum interoperability, consider adding users of the Project Collection Administrators group to the following groups in SharePoint Products:

• Site Collection Administrators group for the site collection that the team project collection uses

## To grant administrative permissions for a team project collection in Reporting Services

1. Start Internet Explorer.

#### ⊠Note

Even if you are logged on with administrative credentials, you must start Internet Explorer as an administrator to perform this function on a computer that is running Windows Server 2008 or Windows Vista. To start Internet Explorer as an administrator, click Start, click All Programs, rightclick Internet Explorer, and then click Run as administrator. For more information, see this topic on the Microsoft Web site: <u>User Account</u> <u>Control</u>.  In the Address bar, specify the following URL, where ReportServer is the name of the server that is running Reporting Services: http://ReportServer/Reports/Pages/Folder.aspx

#### **▲Important**

If you are using a named instance, you must include its name in the path of the reports. You use the following syntax, where ReportServer is the name of the report server for Team Foundation and InstanceName is the name of the instance of SQL Server:

http://ReportServer/Reports\_InstanceName/Pages/Folder.aspx

- 3. On the Home page, choose Folder Settings.
- 4. On the Security page, choose New Role Assignment.
- 5. On the New Role Assignment page, in Group or user name, specify the name of the account for the user or group to whom you want grant administrative permissions.
- 6. In Role, select the Team Foundation Content Manager check box, and then choose OK.

# **See Also**

#### Tasks

Add Users to Team Projects Open the Team Foundation Administration Console Set Administrator Permissions for Team Foundation Server

#### Concepts

<u>Team Foundation Server Permissions</u> Understanding SQL Server and SQL Server Reporting Services

#### **Other Resources**

Interactions Between SharePoint Products and Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Set Administrator Permissions for Team Foundation Server

See Also Send Feedback

You can grant administrator permissions to users in Visual Studio Team Foundation Server by adding them to the Team Foundation Administrators group in Team Foundation Server. If your deployment uses resources in SQL Server Reporting Services, you must also grant those users permissions in that program. Unlike previous versions of Team Foundation Server, you do not have to grant administrators of Team Foundation permissions in SharePoint Products. The permissions that are required for SharePoint Products interoperability are granted to the service account for Team Foundation Server. However, for optimum interoperability and ease of use, you might want to grant members of the Team Foundation Administrators group membership in certain groups in SharePoint Products.

Administrators maintain at least one server that is running Team Foundation Server, and they administer permissions and security for other roles at the server level and at the level of team project collections. Members of the Team Foundation Administrators group have the highest set of permissions of any users in Team Foundation Server. For most organizations that use Team Foundation Server, administrators create and manage team project collections, in addition to performing any operations that are required to maintain the server.

For information about how to set permissions for other members of the team, see <u>Add Users to Team Projects</u> and <u>Set Administrator Permissions for Team Project</u> <u>Collections</u>.

An administrator for Team Foundation Server must be a member of the following groups or have the following permissions:

• Team Foundation Server: Team Foundation Administrators or have the appropriate server-level permissions set to Allow. For more information,

see Team Foundation Server Default Groups, Permissions, and Roles.

- Windows: Administrators group on the server that is running the administration console for Team Foundation. The administration console requires administrative permissions to operate correctly.
- SharePoint Products: the appropriate groups or permissions in SharePoint Central Administration. Depending on your deployment configuration and security requirements, you might not need to add the user to any groups in SharePoint Products. For optimum interoperability, consider adding them to the following SharePoint Products groups:
  - Farm Administrators
  - Site Collection Administrators group for all site collections that are used by the deployment of Team Foundation Server.

For more information, see Interactions Between SharePoint Products and Team Foundation Server and <u>Service Accounts and Dependencies in Team</u> <u>Foundation Server</u>.

- Reporting Services: Team Foundation Content Manager and either sysadmin or the db\_owner group membership for the configuration database, the reporting and analysis databases, and the databases for team project collections.
- SQL Server: serveradmin and sysadmin for all TFS databases.

#### ⊠Note

Even users who have appropriate permissions might not be able to view team project portals or reports correctly until they add the sites to their Trusted Sites in Internet Explorer. For more information, see this topic on the Microsoft Web site: <u>User Account Control</u>.

You can grant administrative permissions for Team Foundation Server in two ways: from the administrative console or directly through each program for which you want to grant permissions. Granting permissions through the administrative console is simpler but has some requirements. Consider using the administrative console when all of the following conditions are true:

- Your deployment of Team Foundation Server is in a trusted environment where the service account for Team Foundation Server has permissions in SharePoint Products and SQL Server Reporting Services.
- All programs are running on the same computer (a single-server deployment).
- The security requirements for your deployment do not restrict granting one or more of the permissions in the next bulleted list.

By default, adding users from the administration console grants them membership in the following groups in a single-server deployment of Team Foundation Server:

- Team Foundation Administrators group in Team Foundation Server
- The IIS\_IUSRS and TFS\_APPTIER\_SERVICE\_WPG groups in Internet Information Services (IIS)
- The Content Manager role in SQL Server Reporting Services, if reporting is configured
- The Farm Administrators group in SharePoint Products, if the deployment is configured to use SharePT
- The DBO role and TFSExecRole for all databases that Team Foundation Server uses, including collection databases

### **▲Important**

You cannot add a user to the local Administrators group by adding that user's account as a console user. You must manually add the user to that group before that user will have all the permissions that are required to open and use the console. In addition, if you want the user to have sufficient permissions to create a database as part of creating a team project collection, you must grant that user membership in the serveradmin role in SQL Server.

Granting permissions directly in each program in your deployment of Team Foundation Server is more time-consuming, but you can precisely configure the exact permissions that you want to grant to a user. Consider granting permissions directly in each program when any of the following conditions are true:

- Your deployment of Team Foundation Server is a multiple-server deployment.
- Your deployment is in an environment that has security restrictions between Team Foundation Server and the servers that are running SQL Server and SharePoint Products.
- You want to configure different group memberships and permissions levels in SharePoint Products, SQL Server Reporting Services, and Team Foundation Server than those that are automatically granted from the administrative console.

#### **Required Permissions**

To perform these procedures, you must belong to the following groups or have the following permissions:

- Team Foundation Administrators group or have the View instance-level information and Edit instance-level information permissions set to Allow.
- If you want to add permissions for SQL Server Reporting Services, the Team Foundation Content Managers group or the System Administrators group.
- If you want to add permissions for SharePoint Products, the Farm Administrators group, the administrators group for the Web application that supports Team Foundation Server, or the SharePoint Administration group. Group membership will depend on the security architecture of your deployment and the group or groups to which you want to add the user.
- If you want to add role membership for SQL Server, the sysadmin role on each server that hosts databases for Team Foundation Server

#### **▲Important**

To perform administrative tasks such as creating team project collections, your user account requires administrative permissions, and the service account that the Team Foundation Background Job Agent uses also must have certain permissions granted to it. For more information, see <u>Service Accounts and</u> <u>Dependencies in Team Foundation Server</u> and <u>Team Foundation Background</u> <u>Job Agent</u>.

For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

# To grant administrative permissions in a trusted environment from the administration console for Team Foundation

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the server, and then choose Application Tier.
- 3. In the Administrative Console Users section, choose Add.
- 4. In Add Team Foundation Server Administration Console User, specify the user account that you want to add as a member of Team Foundation Administrators.

If you are not sure of the alias, choose Search. You can also expand the Advanced Features section and modify the selections for Add required permissions to create collections and Add required permissions to change service accounts as best suits your operational needs and the security requirements for the user whom you are adding.

- 5. When you are satisfied with your selections, choose OK.
- 6. Review the progress information in the Add Administration Console User window.
- 7. (Optional)When the process completes, choose the link to open the log file.

8. Choose Close to return to the administration console.

#### **▲Important**

You cannot add a user to the local Administrators group by adding that user's account as a console user. You must manually add the user to that group before that user will have all the permissions required to open and use the console. In addition, if you want the user to have permissions sufficient to create a database as part of creating a team project collection, you must grant that user membership in the serveradmin role in SQL Server.

### To grant administrative permissions in Team Foundation Server

1. Open the administration console for Team Foundation.

For more information, see <u>Open the Team Foundation Administration</u> <u>Console</u>.

- 2. Expand the server, and then choose Application Tier.
- 3. In the Administrative Console Users section, choose Add.
- 4. In Add Team Foundation Server Administration Console User, specify the user account that you want to add as a member of Team Foundation Administrators.

If you are not sure of the alias, choose Search.

You can also expand the Advanced Features section and modify the selections for Add required permissions to create collections and Add required permissions to change service accounts as best suits your operational needs and the security requirements for the user whom you are adding. By default, leave these check boxes selected.

- 5. When you are satisfied with your selections, choose OK.
- 6. Review the progress information in the Add Administration Console User window.

- 7. (Optional) When the process completes, choose the link to open the log file.
- 8. Choose Close to return to the administration console.

#### **▲Important**

You cannot add a user to the local Administrators group by adding that user's account as a console user. You must manually add the user to that group before that user will have all the permissions required to open and use the console. In addition, if you want the user to have permissions sufficient to create a database as part of creating a team project collection, you must grant that user membership in the serveradmin role in SQL Server.

# To grant administrative permissions in SharePoint Foundation 2010

1. On the server that is running SharePoint Products, open SharePoint Central Administration.

For more information, see Access Site Administration or Central Administration for SharePoint Products.

2. Grant permissions that are appropriate for this user at the farm or the Web application level, depending on your security needs.

For more information, see Windows SharePoint Services Roles and the following page on the Microsoft Web site: <u>SharePoint Products Tech</u> <u>Center</u>. For optimum interoperability, consider adding users of the Team Foundation Administrators group to the following groups in SharePoint Products:

- Farm Administrators
- Site Collection Administrators group for all site collections that the deployment of Team Foundation Server uses

# To grant administrative permissions in Reporting Services

1. Start Internet Explorer.

#### ⊠Note

Even if you are logged on with administrative credentials, you must start Internet Explorer as an administrator to perform this function on a computer that is running Windows Server 2008 or Windows Vista. To start Internet Explorer as an administrator, choose Start, choose All Programs, open the sub-menu for Internet Explorer, and then choose Run as administrator. For more information, see this topic on the Microsoft Web site: <u>User Account Control</u>.

 In the Address bar, specify the following URL, where ReportServer is the name of the server that is running Reporting Services: http://ReportServer/Reports/Pages/Folder.aspx

#### **▲Important**

If you are using a named instance, you must include its name in the path of the reports. You use the following syntax, where ReportServer is the name of the report server for Team Foundation and InstanceName is the name of the instance of SQL Server: http://ReportServer/Reports\_InstanceName/Pages/Folder.aspx

- 3. On the Home page, choose Folder Settings.
- 4. On the Security page, choose New Role Assignment.
- 5. On the New Role Assignment page, in Group or user name, specify the account name of the user or group to whom you want grant administrative permissions.
- 6. In Role, select the Team Foundation Content Manager check box, and then choose OK.

# See Also

#### Tasks

Add Users to Team Projects

### Concepts

<u>Team Foundation Server Permissions</u> <u>Team Foundation Server Default Groups, Permissions, and Roles</u> <u>Configuring Users, Groups, and Permissions</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Managing Permissions
Send Feedback

You can add, change, or remove permissions for users and groups in Visual Studio Team Foundation Server (TFS). Permissions in TFS can be explicitly set, and they can also be inherited depending on a user's membership in specific groups at each of the four levels of permissions: the team level, the project level, the collection level, and the server level. You can view the explicit and implicit permissions for any user or group in TFS, and you can change permissions for users and groups at everything except the team level. Team-level permissions are determined by membership within the team, and cannot be manually set or changed.

When you add a user or group to TFS, you might also need to add that user or group to two other components on which your deployment might depend: SharePoint Products and SQL Server Reporting Services. If your deployment is configured with these resources, you must add users and groups to those programs and grant the appropriate permissions for those users or groups before all operations will function correctly in TFS.

Because of this complexity, it can be difficult to manage individual users and their associated permissions in deployments of TFS. It is much simpler to use Active Directory to organize users into role-based groups and then add each group to TFS, SharePoint Products, and Reporting Services with the appropriate permissions. By taking this approach, you can manage only a few groups across these three programs, instead of many individual users. You can add users to Active Directory groups as needed without having to change that group membership or permissions within those three programs. You can find an example of this here: Configure Team Foundation Server to Support Your Development Teams.

As an administrator, you control what tasks users can perform by specifying

group membership and permissions. To simplify this task, Team Foundation provides default groups and permissions settings. You can use the default groups and settings as they are, customize them, or create your own groups. The topics in this section provide details about permissions.

In Visual Studio Team Foundation Server 2012, you can use the new features in Team Web Access to view and manage the permissions for users and groups at the project and collection level. This includes the ability to quickly view both inherited and explicitly set permissions, as well as view a user's membership in groups:

Control Panel > DefaultCollection > Phone Saver				
overview iterations areas	security alerts			
groups users	NORTHAMERICA >	Jill Frank (Fabrikam)		
Search	permissions member of			
Adam Barr (Fabrikam)	Create test runs Delete team project	not set		
🔲 Jill Frank (Fabrikam)	Delete test runs	not set		
🖳 Julia Ilyina (Fabrikam)	Manage test configurations Manage test environments View project-level information View test runs clear explicit permissions	not set not set inherited allow not set		
	Save Changes Undo Chang	ges		

You must also configure access to Team Web Access features in the administration context of Team Web Access:

Control Panel			
control panel web access	extensions		
Export Audit Log	Web Access Permissions		
Limited	News		
Standard (Default)	Name Standard		
Full	Standard Features		
	Set as default web access   Add   🗿   Search		
	Display Name Username or Scope		
	No identities found in current scope.		

You must use the administration console for Team Foundation to view and manage server-level permissions for users, groups, and service accounts.

# In This Section

#### Team Foundation Server Permissions

Describes the permissions model for TFS as well as all of the individual permissions that you can set for users and groups.

#### Team Foundation Server Default Groups, Permissions, and Roles

Describes the groups and permissions that are created by default when you install TFS or create a team project, including the permissions that are assigned to those groups by default.

#### Change Permissions for a Group or User

Describes how to change the permissions for a group in TFS or an individual user account that has been added directly to TFS.

#### View Permissions

Describes how to view permissions for users and groups at a server or project level.

#### Add Users to Team Projects

Describes how to add users to default groups in TFS, SharePoint Products, and Reporting Services in order to grant those users the permissions that they need to act as readers, contributors, or administrators within that project.

#### Set Administrator Permissions for Team Project Collections

Describes how to add users to default groups in TFS, SharePoint Products, and Reporting Services in order to grant those users the permissions that they need to administer a team project collection.

Set Administrator Permissions for Team Foundation Server

Describes how to add users to default groups in TFS, SharePoint Products, and Reporting Services in order to grant those users the permissions that they need to administer TFS.

#### Change access levels

Describes how to set the default level of features available in Team Web Access, how to add and remove groups to levels of features, and how to audit usage.

# **Related Sections**

Configure Team Foundation Server to Support Your Development Teams

Collaborate (dig deeper)

Configuring Users, Groups, and Permissions

Team Foundation Server Services and Service Accounts

How to: Control Access to Team Foundation Version Control

Configuring Your Server Using the Team Foundation Administration Console

Organizing Your Server with Team Project Collections

Team Foundation Server Architecture

Team Foundation Server Concepts

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Team Foundation Server Permissions
See Also Send Feedback

You can control access to the team projects and team project collections in your deployment of Visual Studio Team Foundation Server (TFS) by understanding and configuring the permissions that are assigned to the users and groups in that deployment. Team Foundation Server includes several utilities to help you view and manage permissions. For individual users and groups, use Team Web Access by following the procedures in View or change your permissions or join another team, or use Team Explorer by following the procedures in <u>Change Permissions</u> for a Group or User. You can also use these tools to deny users access to team projects. For more information, see Restricting access to projects in the deployment.

This is a reference topic about all the permissions available in a TFS deployment and what they do. For step-by-step examples of how to create custom groups, configure permissions to control access to resources, and other options, see Restrict access in TFS.

#### ⊠Note

This topic does not discuss the permissions for SharePoint Products or SQL Server Reporting Services. This topic discusses only those permissions that you set in TFS. For more information about how to set permissions in Reporting Services and SharePoint Products for users in TFS, see <u>Add Users to Team</u> <u>Projects, Set Administrator Permissions for Team Project Collections</u>, and <u>Set</u> <u>Administrator Permissions for Team Foundation Server</u>.

In this topic

• What's new in permissions

- Overview of default permissions groups
- <u>Permission states</u>
  - <u>Deny</u>
  - <u>Allow</u>
  - <u>Not set</u>
- <u>Inheritance</u>
- <u>Setting Permissions</u>
  - <u>Server-level permissions</u>
  - <u>Collection-level permissions</u>
  - <u>Project-level permissions</u>
  - <u>Build-level permissions</u>
  - Work item query permissions
  - Area-level permissions for work item tracking
  - Iteration-level permissions for work item tracking
  - <u>Version control permissions</u>
  - Lab Management permissions

# **What's new in permissions**

In this release of Team Foundation Server, the management interface for permissions has largely shifted to Team Web Access. You can open Team Web Access in the administration context and view and manage permissions at every level except for the server level. In addition, you can view information about why and how an individual permission has been set for a user or group by hovering over that permission and then choosing the Why? option.

This release of Team Foundation Server also creates a default team group when you create a team project. Although team groups have no manually configurable permissions themselves, membership in this group is required for users to use the Team features in Team Web Access. Additionally, this default team group is automatically added to the Contributors default group for the project. Unless you make modifications, any user you add to the team will have all the permissions attributed to the Contributors group.

Team Web Access has new controls for displaying the features that are available to users, and membership in the access groups for Team Web Access controls whether a user can see and use these controls. You can set access levels for groups and users, as well as set the default level for the entire deployment of TFS. For more information, see

Change access levels.

# **Overview of default permission groups**

Permissions determine the authorization for user actions such as workspace administration and project creation. When you create a project in TFS, five default groups are created for that project regardless of your choice of process template. By default, each of these groups has a set of permissions that are defined for them and that govern what members of those groups are authorized to do.

- Project Administrators
- Contributors
- Readers
- Build Administrators
- The default team group, which has the name of your project appended by Team. For example, if you create a team project named "Code Sample," the group name for the team will be "Code Sample Team."

In addition to the default groups that are created for each team project, when you create a team project collection, seven default groups are created for that collection regardless of your choice of process template. Each of those groups also has a set of permissions that are defined for them.

- Project Collection Administrators
- Project Collection Service Accounts
- Project Collection Build Administrators
- Project Collection Build Service Accounts
- Project Collection Valid Users

- Project Collection Proxy Service Accounts
- Project Collection Test Service Accounts

Four default groups are created at the server level when you install Team Foundation Server. Each of these groups has a set of permissions that are defined for them.

- Team Foundation Administrators
- Team Foundation Service Accounts
- Team Foundation Valid Users
- SharePoint Web Application Services

To effectively manage user membership in these default groups and to create custom groups, administrators must first understand the meaning of the permissions and the security implications for explicitly setting permissions. For more information about which permissions are assigned by default to each group, see

Team Foundation Server Default Groups, Permissions, and Roles. **Tip** 

Certain features in TFS are only available to users who have the appropriate licensing level for those features. Access to those features is not controlled by permissions but by membership in licensing groups for Team Web Access. For more information, see <u>Add Users to Team Projects</u>, <u>Change access levels</u>, and the licensing whitepaper for Team Foundation Server.

# **Permission states**

You can specify two explicit authorization states for permissions in Team Foundation Server: Deny and Allow. There is also an implicit authorization that neither sets the permission to Allow nor sets the permission to Deny. This authorization is an implicit Deny state that is referred to as Not set.

### **Deny**

Deny denies authorization for the user or group to perform the actions that are stated in the permission description. Deny is the most powerful permission state in TFS. If a user belongs to a TFS group that has a specific permission set to deny, that user cannot perform that function, even if he or she belongs to another group that has that permission set to allow. The only exception to this rule occurs when the user is a member of the Project Collection Administrators group, unless otherwise stated in the description of the permission. By default, if a user is a member of an administrators group, the permissions of that group override an explicit deny for that user in Team Foundation Server. If a permission setting overrides the default administrative permission, that override is noted in the description of the permission setting is noted in the description of the permission setting overrides the default administrative permission, that override is noted in the description of the permission setting is noted in the description of the permission setting overrides the default administrative permission.

### Allow

Allow grants authorization for the user or group to perform the actions that are stated in the permission description. Allow is the second-most powerful permission state in TFS and is set most frequently, both explicitly and through inheritance to inherited allow. If you do not explicitly set a permission to allow, or membership in a default group does not set a permission to inherited allow, a user or group cannot perform that action in Team Foundation Server.

### **■Not set**

By default, most permissions in TFS are not set to either deny or allow. The permissions are left not set, which implicitly denies both users and groups authorization to perform the actions that are specified in the permission
description. However, because the permission is neither explicitly set to deny nor explicitly set to allow, authorization for that permission can be inherited from other groups of which the user or group is a member.

# Inheritance

When permission is not set for a user or group, the user or group can be affected by the explicit state for the permission for groups to which they belong because permissions in TFS are inherited. For example, when you review the permissions for a user or group, you might see both allow and inherited allow set for permissions. The latter permission is inherited from some other group the user or group belongs to. In this example, a user might belong to a group at the project level and a group at the collection level in a project. If one of those groups has a permission not set, the user will have the inherited allow permission to perform the actions that are controlled by that permission. The user inherits permissions from both groups, and the allow permission takes precedence over the not set permission.

To understand why a permission is inherited, you can hover over the permission setting, and then choose Why?. A new window will open that displays the inheritance information for that permission.

Control Panel > DefaultCollection > Phone Saver			
overview iterations areas	security alerts		
groups users	NORTHAMERICA >	Jill Frank (Fabrikam)	
Search D Adam Barr (Fabrikam) Annie Herriman	Create test runs Delete team project Delete test runs	not set	
Jill Frank (Fabrikam)	Edit project-level information	inherited allow   Why?	
Julia Ilyina (Fabrikam)	Manage test configurations Manage test environments View project-level information View test runs clear explicit permissions Save Changes Undo Chang	not set not set inherited allow not set	

#### ⊠Note

Permissions that are set outside TFS, such as in SharePoint Products, are not inherited in TFS. They are not discussed in this topic.

Certain authorization settings take precedence over other authorization settings. In TFS, the deny permission takes precedence over all other permission settings, including allow, for that explicit structure. The deny permission does not take precedence if it is inherited from a hierarchical parent, such as in version control. For example, a user might belong to two groups in a project. For one group, Publish test results permission is set to deny; the other group has that permission set to allow. The deny setting takes precedence and the user is not authorized to publish test results. The only exceptions to this rule occur when either the explicit deny is inherited from a hierarchical parent or the user is a member of one of the following groups:

- Project Administrators
- Project Collection Administrators

• Team Foundation Administrators

In hierarchical structures such as version control and work item tracking, the explicit permissions that are set on a particular object override those that are inherited from the parent objects.

# **Setting Permissions**

Many of the permissions that you might want to set for TFS are controlled through the Team Web Access user interface, which you can open directly through Team Web Access, or you can open from Team Explorer. You can set permissions on a collection basis (collection-level permissions), on a project basis (project-level permissions), or by adding a user to a team group, a team basis. (Team-level permissions are implicit in membership for a team and not directly configurable.) You can also set area-level and iteration-level permissions for viewing and interacting with work items on a project basis. For more information about how to set permissions for users and groups, see Collaborate (dig deeper) and

### Add Users to Team Projects.

To set permissions on a server basis (server-level permissions), you cannot use Team Web Access. You must use the administration console for Team Foundation Server to set server-level permissions.

## **Server-level permissions**

Server-level permissions are not specific to a single team project or team project collection. They are set on a deployment-wide basis, and they grant permissions that can affect every project and collection in the deployment.

You can set these permissions for only two categories of users:

- Server-level users and groups, such as Team Foundation Administrators
- Custom groups that you create and add to the server level

You can set these permissions by opening the administration console for Team Foundation. You can also set these permissions by using the **TFSSecurity** command-line tool. For more information, see

<u>Configuring Your Server Using the Team Foundation Administration Console</u> and <u>Changing Groups and Permissions with TFSSecurity</u>. The following table lists each server-level permission and provides a brief description of its purpose.

	Permission Name	Name at command line	Description
	Administer warehouse	ADMINISTER_WAREHOUSE	Users who have this permission can change warehouse settings by using the ChangeSetting Web method of the WarehouseController.asmx Web service. For example, you could allow users to set the update interval for calculating the OLAP cubes.
	Create team project collection	CreateCollection	Users who have this permission can create and administer team project collections in Team Foundation Server.
	Delete		Users who have this permission can delete a team project collection from the deployment.
team project collection	team	DeleteCollection	⊠Note
		Deleting a team project collection will not delete the database for the collection from SQL	

# Edit GENERIC\_WRITE instancelevel tf: AdminConfiguration information tf: AdminConnections

Make requests on behalf of others

### Server.

Users who have this permission can edit serverlevel permissions for users and groups in Team Foundation Server. They can add or remove serverlevel application groups from the collection. When set through the menus, the Edit instance-level information permission also implicitly allows the user to modify version control permissions. To grant all these permissions at a command prompt, you must use the tf.exe Permission command to grant the AdminConfiguration and AdminConnections permissions in addition to GENERIC\_WRITE.

Note Default server-level groups such as Team Foundation Administrators cannot be removed.

Users who have this permission can perform operations on behalf of other users or services. This permission should be assigned only to service

		accounts.
Trigger Events	TRIGGER_EVENT	Users who have this permission can trigger alert events within Team Foundation Server. This permission should be assigned only to service accounts and members of the Team Foundation Administrators group.
Use full Web Access features	FullAccess	Users who have this permission can use all of the features of Team Web Access. If this permission is set to Deny, the user will only see those features permitted for the Limited group in Team Web Access (see <u>Change access</u> levels). A Deny will override any implicit allow, even for accounts that are members of administrative groups such as Team Foundation Administrators.
View instance- level information	GENERIC_READ	Users who have this permission can view server-level group membership and the permissions of those users.

# **Collection-level permissions**

Collection-level permissions are not specific to a single project. Instead, they are set on a collection-wide basis. You can set these permissions for only three categories of users:

- Collection-level users and groups, such as Project Collection Administrators
- Project-level groups that have been added to the collection level on your server that is running Team Foundation
- Custom groups that you create and add to the collection level

You can set these permissions by choosing the collection in Team Explorer by choosing the Team menu, choosing Team Project Collection Settings, and then choosing Security, or by opening Team Web Access in administration mode, navigating to the collection level, and then choosing the Security tab. You can also set these permissions by using the **TFSSecurity** command-line tool, except for those command-line tools that have a **tf**: designation. For those tools with the **tf**: designation, use the **Permission** command of the **tf** command-line utility for version control to set the permissions. For more information, see

Changing Groups and Permissions with TFSSecurity and Permission Command.

Permission Name	Name at Command Line	Description
Administer build resource permissions		Users who have this permission can administer build resources.
Administer		Users who have this permission can configure the integration of Team

Project Server integration	AdministerProjectServer	Foundation Server with Project Server to support synchronization between the two server products.
Administer shelved changes	tf: AdminShelvesets	Users who have this permission can delete shelvesets created by other users.
Administer workspaces	tf: AdminWorkspaces	Users who have this permission can create workspaces for other users and delete workspaces created by other users.
Alter trace settings	DIAGNOSTIC_TRACE	Users who have this permission can change the trace settings for gathering more detailed diagnostic information about Web services for Team Foundation Server.
Create a workspace	tf: CreateWorkspace	Users who have this permission can create a version control workspace.

		U pe pi pi
Create new projects	CREATE_PROJECTS	Y h b S a s c V V ir C P
		U pe te co
Delete team project	Delete	E p d w c d p re

Jsers who have this permission can create projects in the team project collection.

### ⊠Note

You must not only have this permission but also run Visual Studio as an administrator to successfully complete the Create a New Team Project Wizard. For more information, see Create a Team Project.

Users who have this permission can delete team projects in the team project collection.

### **AImportant**

Deleting a team project will delete all data that is associated with the project. You cannot undo the deletion of a team project except by restoring the collection to a point

before the project was deleted.

Users who have this permission can edit collection-level permissions for users and groups in the team project collection. They can add or remove collection-level Team Foundation Server application groups from the collection. When set through the menus, the Edit collection-level information permission also implicitly allows the user to modify version control permissions. To grant all these permissions at a command prompt, you must use the tf.exe Permission command to grant the AdminConfiguration and **AdminConnections** permissions, in addition to GENERIC WRITE.

ĭNote

# GENERIC\_WRITE

Edit collectionlevel tf: AdminConfiguration information tf: AdminConnections

		Default collection- level groups such as Project Collection Administrators cannot be removed.
Make requests on behalf of others	Impersonate	Users who have this permission can perform operations on behalf of other users or services. This permission should be assigned only to service accounts.
Manage build resources	ManageBuildResources	Users who have this permission can manage the build computers, build agents, and build controllers for the team project collection. These users can also grant or deny the View build resources and Use build resources permissions for other users.
Manage process template	MANAGE_TEMPLATE	Users who have this permission can download, create, edit, and upload process templates to the team project

		collection.
Manage test controllers	MANAGE_TEST_CONTROLLERS	Users who have this permission can register and de- register test controllers for the team project collection.
Manage work item link types	WORK_ITEM_WRITE	Users who have this permission can add, remove, and change the types of links for work items.
Trigger events	TRIGGER_EVENT	Users who have this permission can trigger project alert events within the team project collection. This permission should be assigned only to service accounts.
Use build resources	UseBuildResources	Users who have this permission can reserve and allocate build agents. This permission should be assigned only to service accounts for build services.

View build resources	ViewBuildResources	permission can view build controllers and build agents that are configured for the collection. To use those resources, you need additional permissions.
View collection-level information	GENERIC_READ	Users who have this permission can view collection-level group membership and the permissions of those users.
View system synchronizatior information	N SYNCHRONIZE_READ	Users who have this permission can call the synchronization application programming interfaces. This permission should be assigned only to service accounts.

Users who have this

# **Project-level permissions**

Project-level permissions are specific to a single project's users and groups. You can set these permissions by choosing the project in Team Explorer, choosing Settings, and then choosing Security, or by opening Team Web Access in administration mode, navigating to the project, and then choosing the Security tab. You can also set these permissions by using the TFSSecurity command-line tool.

Permission Name	Name at Command Line	Description
Create test runs	PUBLISH_TEST_RESULTS	Users who have this permission can add and remove test results and add or modify test runs for the team project.
Delete team project	DELETE	Users who have this permission can delete the project for which they have this permission from Team Foundation Server.
Delete test runs	DELETE_TEST_RESULTS	Users who have this permission can delete a scheduled test for this team project.

Edit project- level information	GENERIC_WRITE	Users who have this permission can edit project-level permissions for users and groups on Team Foundation Server.
Manage test configurations	MANAGE_TEST_CONFIGURATIONS	Users who have this permission can create and delete test configurations for this team project.
Manage test environments	MANAGE_TEST_ENVIRONMENTS	Users who have this permission can create and delete test environments for this team project.
View project- level	GENERIC_READ	Users who have this permission can view project-level group

information	membership and the permissions of those project users.
View test runs VIEW_TEST_RESULTS	Users who have this permission can view test plans in this node.

# **Build-level permissions**

Build-level permissions are specific to a single project's users and groups. You can set build permissions at the team project level, and you can also set permissions for specific build definitions. You can set these permissions by opening the project in Team Explorer, choosing Builds, and under Actions, choosing Security. You can apply permissions to a specific build definition by choosing the build definition and then choosing Security. If you want to apply permissions to a build folder, select that folder, and from its sub-menu, choose Security. Additionally, you can set these permissions by using the TFSSecurity command-line tool.

Permission Name	Name at Command Line	Description
Administer build		Users who have this permission can administer the
permissions		build permissions

	for other users.
View builds ViewBuilds	Users who have this permission can view the queued and completed builds for this team project.
Edit build quality EditBuildQuality	Users who have this permission can add information about the quality of the build through the user interface for Team Foundation Build.
Retain indefinitely	Users who have this permission can mark a build so that it will not be automatically deleted by

gh

it V any applicable

		retention policy.
Delete builds	DeleteBuilds	Users who have this permission can delete a completed build.
Manage build qualities	ManageBuildQualities	Users who have this permission can add or remove build qualities.
Destroy builds	DestroyBuilds	Users who have this permission can permanently delete a completed build.
Update build	UpdateBuildInformation	Users who have this permission can add build information nodes to the system, and can also add information

information		about the quality of a build. This permission should be assigned only to service accounts.
Queue builds	QueueBuilds	Users who have this permission can put a build in the queue through the interface for Team Foundation Build or at a command prompt. They can also stop the builds that they have queued.
Manage build queue	ManageBuildQueue	Users who have this permission can cancel, re-prioritize, or postpone queued builds.
		Users who

Stop builds	StopBuilds	have this permission can stop any build that is in progress, including builds queued and started by another user.
View build definition	ViewBuildDefinition	Users who have this permission can view the build definitions that have been created for the team project.
Edit build definition	EditBuildDefinition	Users who have this permission can create and modify build definitions for this project.
Delete build definition	DeleteBuildDefinition	Users who have this permission can delete build definitions for this project.

Override check-in validation by build	OverrideBuildCheckInValidation	Users who have this permission can commit a changeset that affects a gated build definition without triggering the system to shelve and build their changes first. This permission should be assigned only to service accounts for build services and to build administrators who are responsible for the quality of the code. For more information, see Check in to a Folder that is Controlled by a Gated Check-in Build Process.	
--	--------------------------------	---	--

### **Work item query permissions**

Work item query permissions are specific to the queries and query folders that you create. You can set permissions on queries and folders that are created under Team Queries to enable or restrict access. You can set these permissions by choosing the project in Team Explorer, opening Work Items, opening the submenu for the query or folder for which you want to set permissions, and then choosing Security. You can also set these permissions on the query in Team Web Access. Additionally, you can set these permissions by using the TFSSecurity command-line tool.

Name	Name at Command Line	Description
Contribute	CONTRIBUTE	Users who have this permission can view and modify this query or query folder.
Delete	DELETE	Users who have this permission can delete a query or query folder and its contents.
		Users who have this

### Permission ...

Manage Permissions	MANAGEPERMISSIONS	permission can manage the permissions for this query or query folder.
Read	READ	Users who have this permission can view and use the query or the queries in a folder, but cannot modify the query or query folder contents.

## ■Area-level permissions for work item tracking

Area-level permissions are specific to a single project's users and groups. You can set these permissions by choosing the project in Team Explorer, choosing Settings, choosing Work Item Areas, choosing the down arrow, and then choosing Security, or by opening Team Web Access in the administration context, navigating to the project level, choosing the Areas tab, choosing the down arrow, and then choosing Security. Additionally, you can set these permissions by using the TFSSecurity command-line tool.

### ⊠Note

Some work item tracking operations require multiple permissions. For example, you need multiple permissions to delete a node.

Permission Name	Name at Command Line	Description
Create child nodes	CREATE_CHILDREN	Users who have this permission can create area nodes. Users who have both this permission and the Edit this node permission can move or re- order any child area nodes.
Delete this node	DELETE	Users who have both this permission and the Edit this node permission for another node can delete area nodes and reclassify existing work items from the deleted node. If the deleted node has child nodes, those nodes are also deleted.

Edit this node	GENERIC_WRITE	Users who have this permission can set permissions for this node and rename area nodes.
Edit work items in this node	WORK_ITEM_WRITE	Users who have this permission can edit work items in this area node.
Manage test plans	MANAGE_TEST_PLANS	Users who have this permission can create and edit test plans for this node. If test plans have not been run, you can also delete them.
View permissions for this node	GENERIC_READ	Users who have this permission can view the security settings for this node.

View work items in this node	WORK_ITEM_READ	have this permission can view, but not change, work items in this area node. If this permission is set to Deny, the user will not be able to see any work items in this area node. A Deny will override any implicit allow, even for accounts that are members of administrative groups such as Team Foundation Administrators.
------------------------------------	----------------	--

# **Iteration-level permissions for work item tracking**

Iteration-level permissions are specific to a single project's users and groups. You can set these permissions by choosing the project in Team Explorer, choosing Settings, choosing Work Item Iterations, choosing the down arrow, and then choosing Security, or by opening Team Web Access in the administration context, navigating to the project level, choosing the Iterations tab, choosing the down arrow, and then choosing Security. Additionally, you can set these permissions by using the TFSSecurity command-line tool.

### ⊠Note

Some work item tracking operations require multiple permissions. For example, you need multiple permissions to delete a node.

Permission Name	Name at Command Line	Description
Create child nodes	CREATE_CHILDREN	Users who have this permission can create iteration nodes. Users who have both this permission and the Edit this node permission can move or re-order any child iteration nodes.
		Users who have both this permission and the Edit this node permission for another node can delete iteration

Delete this node	DELETE	nodes and reclassify existing work items from the deleted node. If the deleted node has child nodes, those nodes are also deleted.
Edit this node	GENERIC_WRITE	Users who have this permission can set permissions for this node and rename iteration nodes.
View permissions for this node	GENERIC_READ	Users who have this permission can view the security settings for this node.

# **Version Control Permissions**

Version control permissions are specific to source code files and folders. You can

set these permissions by opening the sub-menu for the folder or file in Source Control Explorer, and then choosing Security. You can also set these permissions by using the **tf** command-line tool for version control.

In version control permissions, explicit deny takes precedence over administrator group permissions.

Permission Name	Name at Command Line	Description
Read	tf: Read	Users who have this permission can read the contents of a file or folder. If a user has Read permissions for a folder, the user can see the contents of the folder and the properties of the files in it, even if the user does not have permission to open the files.

Check out tf: PendChange	Users who have this permission can check out and make a pending change to items in a folder. Examples of pending changes include adding, editing, renaming, deleting, undeleting, branching, and merging a file. Pending changes must be checked in, so users will also need the Check in permission to share their changes with the team.
--------------------------	--

Check in	tf: Checkin	Users who have this permission can check in items and revise any committed changeset comments. Pending changes are committed at check-in.
Label	tf: Label	Users who have this permission can label items.
Lock	tf: Lock	Users who have this permission can lock and unlock folders or files.
Revise other users' changes	tf: ReviseOther	Users who have this permission can edit the comments on checked- in files, even if

	another user checked in the file.
Unlock other users' tf: UnlockOther changes	Users who have this permission can unlock files locked by other users.
Undo other users' tf: UndoOther changes	Users who have this permission can undo a pending change made by another user.
Administer labels	Users who have this permission can edit or delete labels created by another user.
	Users who have this permission

Manage permissions	tf: AdminProjectRights	can manage other users' permissions for folders and files in version control.
Check in other users' changes	tf: CheckinOther	Users who have this permission can check in changes that were made by other users. Pending changes will be committed at check-in.
Merge	tf: Merge	Users who have this permission for a given path can merge changes into this path.
		Users who have this permission for a given

Manage tf: ManageBranch branch tf: ManageBranch branch Use hav peri- can this only also the peri- for targ Use can trans for targ Use can the peri- for targ	owing ons on a nch: edit perties, parent it, convert a ler. ers who e this mission branch branch branch branch branch y if they have Merge mission the et path. ers not a te nches n a nch for ch they
---	---
do not have the Manage Branch permission.

## **Lab Management permissions**

Visual Studio Lab Management permissions are specific to virtual machines, environments, and other resources. You can set these permissions by using the **TFSLabConfig** command-line tool.

Permission Name	Name at Command Line	Description
View Lab Resources	Read	Users who have this permission can view information for the various Lab Management resources, which include collection host groups, project host groups, and environment. To view information about a specific lab resource, you must have the View Lab Resources permission for that resource.
		Users who have this permission can edit the locations of Lab Management resources,

Manage Lab	Managel ocation
Locations	

which include collection host groups, collection library shares, project host groups, and project library shares. To edit a specific location, you must have the Manage Lab Location permission for that location.

This permission for collection-level locations (collection host groups and collection library shares) also allows you to create project-level locations (project host group and project library share).

Users who have this permission can delete the locations for Lab Management resources, which include collection host groups, collection library shares, project host groups, and project library shares. To delete a location, you must have the Delete Lab Location permission for that location.

#### Delete Lab Locations

DeleteLocation

Write Environment and Virtual Machines

Edit Environment and Virtual Machines

Delete Environment and Virtual Machine

Import Virtual Machine

Create

Users who have this permission for a project host group can create environments. Users who have this permission for a project library share can store environments and templates.

Users who have this permission can edit environments and templates. The permission is checked for the object that is being edited.

Users who have this permission can delete environments and templates. The permission is checked for the object that is being deleted.

Users who have this permission can import a virtual machine from a VMM library share.

This permission differs from Write because it only creates an object

		in Lab Management and does not write anything to the Virtual Machine Manager host group or library share.
Manage Permissions	ManagePermissions	Users who have this permission can modify the permissions for a Lab Management object. This permission is checked for the object whose permissions are being modified.
Manage Child Permissions	ManageChildPermissions	Users who have this permission can change the permissions of all the child Lab Management objects. For example, if a user has Manage Child Permission for a team project host group, the user can change permissions for all the environments under that team project host group.
Start	Start	Users who have this permission can start an environment.

5	Stop	Stop	Users who have this permission can stop an environment.
F	Pause Environment	Pause	Users who have this permission can pause an environment.
N	⁄Ianage Snapshots	ManageSnapshots	Users who have this permission can perform all snapshot management tasks for an environment, which include taking a snapshot, reverting to a snapshot, renaming a snapshot, deleting a snapshot, and reading a snapshot.

## See Also

#### Tasks

**Open the Team Foundation Administration Console** 

#### Concepts

Collaborate (dig deeper) <u>Configuring Users, Groups, and Permissions</u> <u>Configuring Lab Management with TFSLabConfig</u> Defining a Test Plan

#### **Other Resources**

Get Up and Running With a Single-Server Installation [Tutorial] Changing Groups and Permissions with TFSSecurity Team Foundation Version Control Command Reference Managing Permissions Changing Groups and Permissions with TFSSecurity Send feedback on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Team Foundation Server Default Groups, Permissions, and Roles
See Also Send Feedback

Team Foundation Server includes default groups that help you manage users and permissions within the deployment. The product installation creates these groups for you automatically at the server and collection level. Similarly, when you create a project in Visual Studio Team Foundation Server, a number of groups are created for that project, at various levels:

- Four groups in the project itself, at the project-level
- A default team group, which by default is added to the project-level group Contributors
- Area-level and iteration-level groups for the project
- Build-level groups for the team project

If you add Lab Management to your deployment, groups for Lab Management are added.

To customize projects to better suit your business needs, it helps to understand what permissions are assigned to which users and groups, in addition to what permissions you might want to add to any users or groups that you might add at the server level, the collection level, and the project level. Additionally, if you want to closely align users with specific roles and responsibilities, you need to understand how to align those roles with the default groups that are already assigned to the project. As an alternative, you can create groups that associate directly with each of those roles, and you can assign those groups the permissions that are appropriate to the role.

In this topic

- Default groups and permissions
  - <u>Server-level groups and permissions</u>
  - <u>Collection-level groups and permissions</u>
  - Project-level groups and permissions
  - Area-level groups and permissions
  - Iteration-level groups and permissions
  - <u>Version control groups and permissions</u>
  - Build-level groups and permissions
  - Lab Management groups and permissions

# **Default groups and permissions**

Whenever you create a project in Team Foundation Server, groups are created at the project level. By default, each of those groups has certain permissions assigned to them. You can add permissions to these default groups, in addition to any groups or users whom you want to add at the server, collection, or project level.

### **Server-level groups and permissions**

By default, the following groups exist at the server level when you install Team Foundation Server:

- Team Foundation\**Team Foundation Administrators** Members of this group can perform all operations for Team Foundation Server. This group should be restricted to the smallest possible number of users who need total administrative control over Team Foundation Server. By default, this group contains the Local Administrators group (BUILTIN\Administrators) for any server that hosts the application services for Team Foundation. This group also contains the members of the Server\**Team Foundation Service** Accounts group and the members of the \**Project Server Integration** Service Accounts group.
- Team Foundation\**Team Foundation Valid Users** Members of this group have access to Team Foundation Server. This group automatically contains all users and groups that have been added anywhere within Team Foundation Server. You cannot modify the membership of this group.

#### **▲Important**

If you unset or set the View instance-level information permission to Deny for this group, no users will be able to access the deployment.

• Team Foundation\**Team Foundation Service Accounts** Members of this group have service-level permissions for Team Foundation Server. By

default, this group contains the service account that was supplied during installation. This group should contain only service accounts and not user accounts or groups that contain user accounts. By default, this group is a member of Team Foundation Administrators.

- Team Foundation\Project Server Integration Service Accounts Members of this group have service-level permissions for the Project Server deployments that are configured for interoperation with Team Foundation Server. In addition, members of this group have some service-level permissions for Team Foundation Server. This group should contain only service accounts and not user accounts or groups that contain user accounts. By default, this group is a member of Team Foundation Administrators.
- Team Foundation\SharePoint Web Application Services Members of this group have service-level permissions for the SharePoint Web applications that are configured for use with Team Foundation Server, in addition to some service-level permissions for Team Foundation Server. This group should contain only service accounts and not user accounts or groups that contain user accounts. Unlike the Service Accounts group, this group is not a member of Team Foundation Administrators.
- Team Foundation\**Team Foundation Proxy Service Accounts** Members of this group have service-level permissions for Team Foundation Server Proxy, and have some service-level permissions for Team Foundation Server. This group should contain only service accounts and not user accounts or groups that contain user accounts.

By default, these groups have the permissions in the following table. Unless otherwise stated, the permission is set to Allow. For a full description of each permission, see

Permission Name	By default, set for:	Consider adding to:
	Team	Manually added users or groups who might or must

Administer warehouse	Foundation Administrators Team Foundation Service Accounts	change warehouse settings through the WarehouseController.asmx Web service ChangeSetting Web method.
Create team project collection	Team Foundation Administrators Team Foundation Service Accounts	Users or groups who are responsible for managing the overall health and resource availability for the deployment.
Delete team project collection	Team Foundation Administrators Team Foundation Service Accounts	Users or groups who are responsible for managing the overall health and resource availability for the deployment.
Edit instance- level information	Team Foundation Administrators Team Foundation Service Accounts	Users or groups who are responsible for managing the overall health and resource availability for the deployment.

Make requests on behalf of others	Team Foundation Service Accounts SharePoint Web Application Services	This permission should be assigned only to service accounts and groups that contain only service accounts.
Trigger Events	Team Foundation Administrators Team Foundation Service Accounts	Users or groups who are responsible for managing the overall health and resource availability for the deployment.
Use full Web Access features	Team Foundation Administrators Team Foundation Valid Users	Users and groups who must utilize the full range of features that are available in Team Web Access. If you want to restrict users to a read- only view in Team Web Access, set this permission to Deny.
View	Team Foundation Administrators Team Foundation Service	

instance- level information	Accounts SharePoint Web Application Services	All users or groups who interact with Team Foundation Server.
	Team Foundation Valid Users	

## **Collection-level groups and permissions**

By default, the following groups exist at the collection level when you install Team Foundation Server:

- TeamProjectCollectionName\Project Collection
   Administrators Members of this group can perform all operations for the team project collection. This group should be restricted to the smallest possible number of users who need total administrative control over the collection. By default, this group contains the Local Administrators group (BUILTIN\Administrators) for the server where the application-tier services for Team Foundation have been installed. This group also contains the members of the TeamProjectCollectionName\Service Accounts group.
- TeamProjectCollectionName\**Project Collection Valid Users** Members of this group have access to the team project collection in Team Foundation Server. This group automatically contains all users and groups that have been added anywhere within the team project collection. You cannot modify the membership of this group.

#### **▲Important**

Do not unset or set the View collection-level information permission to Deny for this group.

• TeamProjectCollectionName\**Project Collection Service** 

**Accounts** Members of this group have service-level permissions for the collection and for Team Foundation Server. By default, this group contains the service account that was supplied during installation. This group should contain only service accounts and groups that contain only service accounts. By default, this group is a member of Team Foundation Administrators and Team Foundation Service Accounts.

- TeamProjectCollectionName\**Project Collection Build Administrators** Members of this group have build administration permissions for the collection. This group should be restricted to the smallest possible number of users who need total administrative control over build servers and services for this collection.
- TeamProjectCollectionName\**Project Collection Build Service Accounts** Members of this group have build service permissions for the collection. This group should contain only service accounts and groups that contain only service accounts.
- TeamProjectCollectionName\**Project Collection Proxy Service** Accounts Members of this group have proxy service permissions for the collection. This group should contain only service accounts and groups that contain only service accounts.
- TeamProjectCollectionName\**Project Collection Test Service** Accounts Members of this group have test service permissions for the collection. This group should contain only service accounts and groups that contain only service accounts.

By default, these groups have the permissions in the following table. For a full description of each permission, see

PermissionBy default, setNamefor:		Consider adding to:
	Project Collection Administrators	Build administrators

Administer build resource permissions	Project Collection Build Administrators Project Collection Service Accounts	who regularly administer build servers and resources for the collection.
Administer Project Server integration	Project Collection Administrators Project Collection Service Accounts	None.
Administer shelved changes	Project Collection Administrators Project Collection Service Accounts Project Collection Build Service Accounts	Build administrators who delete shelvesets created by other users.
Administer workspaces	Project Collection Administrators Project Collection Service Accounts	Build administrators who create workspaces for other users and delete workspaces created by other users.
		Other server

Alter trace settings	Project Collection Administrators	administrators who might or must change the trace settings for gathering more detailed diagnostic information about Web services for Team Foundation Server.
	Project Collection Administrators	
	Project Collection Build Administrators	
	Project Collection Build Service Accounts	None. All users have
Create a	Project Collection Contributors	this permission as part of being
workspace	Project Collection Proxy Service Accounts	members of the Project Collection Valid Users
	Project Collection Service Accounts	group.
	Project Collection Test Service Accounts	

	Project Collection	
	Valid Users	
Create new projects	Project Collection Administrators	Project administrators who will regularly create projects.
Delete team project	Project Collection Administrators	Users or groups who are responsible for managing the overall health and resource availability for the deployment.
Edit collection- level information	Project Collection Administrators Project Collection Service Accounts	None.
	Project Collection Administrators	
Make requests on behalf of others	Project Collection Service Accounts	None.
oulcis	SharePoint Web	

	Application Services	
Manage build resources	Project Collection Administrators Project Collection Build Administrators Project Collection Build Service Accounts Project Administrators ProjectName\Build Administrators	Manually added users or groups who might or must administer and schedule builds on the build resources in the collection.
Manage process template	Project Collection Administrators	Project administrators and any manually added users or groups, such as process specialists, who might or must create, edit, download, and upload process templates to Team

		Foundation Server.
Manage test controllers	Project Collection Administrators Project Collection Test Service Accounts	None.
Manage work item link types	Project Collection Administrators	None.
Trigger Events	Project Collection Administrators Project Collection Service Accounts	None. Adding this permission to other users has the potential to allow denial- of-service attacks.
Use build resources	Project Collection Administrators Project Collection Build Service Accounts	Manually added users or groups who might or must queue new builds or browse completed builds in the collection.

	Project Collection Administrators	
View build	Project Collection Build Administrators	None.
	Project Collection Build Service Accounts	
	Project Collection Valid Users	
	Project Collection Administrators	
	Project Collection Build Administrators	
	Project Collection Build Service Accounts	
View	Project Collection Service Accounts	
collection-level information	Project Collection Test Service Accounts	None.
	Project Collection Valid Users	
	SharePoint Web Application Services	

	Project Collection Proxy Service Accounts	
View system synchronization information	Project Collection Administrators	None.

### **Project-level groups and permissions**

By default, the following groups exist at the project level:

- ProjectName\**Project Administrators** Members of this group can administer all aspects of the team project, although they cannot create projects.
- ProjectName\**Contributors** Members of this group can contribute to the project, such as by adding, modifying, and deleting code, and by creating and modifying work items. By default, the team group created when you create a team project is added to this group, and any user you add to the team will be a member of this group. In addition, any team you create for a team project will be added to this group by default, unless you choose a different group from the list.
- ProjectName\**Readers** Members of this group can view the project but not modify it.
- ProjectName\**Build Administrators** Members of this group have build permissions for the project. Members can manage test environments, create test runs, and manage builds.
- ProjectName\TeamName**Team** Members of this group can contribute to the project in multiple ways, such as adding, modifying, and deleting code and creating and modifying work items. The default Team group is created when you create a team project, and by default is added to the Contributors group for the team project. Any new teams you create will also have a group created for them and added to the Contributors group.

Besides these project-level groups, two collection-level groups also appear in every project in Team Foundation Server:

TeamProjectCollectionName\Project Collection Administrators

ĭNote

You cannot change the permissions for this collection-level group.

• TeamProjectCollectionName\Project Collection Build Service Accounts

#### **▲Important**

Do not remove or set the View project-level information permission to Deny for this group.

By default, these groups have the permissions in the following table. For a full description of each permission, see

Permission Name	By default, set for:	Consider adding to:
	Project Administrators, Contributors, Builders, Project Collection	
Create test runs	Project Collection Build Service Accounts,	None.

	Project Collection Test Service Accounts, TeamName Team	
Delete team project	Project Administrators, Project Collection Administrators	None.
Delete test runs	Project Administrators, Team Foundation Administrators, Contributors, TeamName Team	Manually added users or groups that might or must terminate test runs that are in progress or delete old test runs.
Edit project- level information	Project Administrators, Project Collection Administrators	None.

B Ph C A Manage test C configurations B A Ph C S A T C S C S C S C S C S C S C S C S C	Collection Administrators, Project Collection Build Service Accounts, Project Collection Test Fervice Accounts, TeamName Team	None.
Manage test environments Pi C A Pi C B A Pi C S A Ta Ta Ta	Project Administrators, Contributors, Builders, Project Collection Administrators, Project Collection Build Service Accounts, Project Collection Test Project Collection Test Project Collection Test Project Collection Test Project Collection Test Project Collection Test	None.

View project- level information	Administrators, Contributors, Readers, Builders, Project Collection Administrators, Project Collection Build Service Accounts, TeamName Team	All manually added users or groups that require access to this project.
View test runs	Project Administrators, Contributors, Readers, Builders, Project Collection Build Service Accounts, Project Collection Administrators, TeamName Team	All manually added users or groups that require access to this project.

## ■Area-level groups and permissions

By default, the following groups exist at the area level:

- ProjectName\**Project Administrators**
- ProjectName\Contributors

- ProjectName\**Readers**
- ProjectName\**Builders**
- ProjectName\TeamName **Team**
- TeamProjectCollectionName\**Project Collection Administrators**
- TeamProjectCollectionName\Project Collection Build Service Accounts
- TeamProjectCollectionName\**Project Collection Test Service Accounts**

By default, these groups have the permissions in the following table. For a full description of each permission, see

Permission Name	By default, set for:	Consider adding to:
Create child nodes	Project Administrators, Project Collection Administrators	None.
Delete this node	Project Administrators, Project Collection Administrators	Any manually added users or groups that might or must delete area nodes.

Edit this node	Project Administrators, Project Collection Administrators	Any manually added users or groups that might or must rename area nodes.
Edit work items in this node	Project Administrators, Contributors, Builders, Project Collection Administrators, Project Collection Build Service Accounts, TeamName Team	Any manually added users or groups that might or must edit work items in this area node.
Manage Test Plans	Project Administrators, Contributors, Builders, Project Collection Administrators, Project Collection Build Service Accounts,	Any manually added users or groups that might or must manage test plans

	TeamName Team	in this area node.
View permissions for this node	Project Administrators, Contributors, Readers, Builders, Project Collection Administrators, Project Collection Build Service Accounts, Project Collection Test Service Accounts, TeamName Team	Any manually added users or groups that might require access to work items in this area node.
View work items in this node	Project Administrators, Contributors, Readers, Builders, Project Collection Administrators, Project Collection Build Service Accounts, Project Collection Test Service Accounts,	Any manually added users or groups that might or must view, but not edit or change, work items in

TeamName	this area
Team	node.

## **■Iteration-level groups and permissions**

By default, the following groups exist at the iteration level:

- ProjectName\**Project Administrators**
- TeamProjectCollectionName\**Project Collection Administrators**

By default, these groups have the permissions in the following table. For a full description of each permission, see

Permission Name	By default, set for:	Consider adding to:
Create child nodes	Project Administrators, Project Collection Administrators	None.
Delete this node	Project Administrators, Project Collection Administrators	Any manually added users or groups that might or must delete iteration nodes.

Edit this node	Project Administrators, Project Collection Administrators	Any manually added users or groups that might or must rename iteration nodes.
View permissions for this node	Project Administrators, Project Collection Administrators	Any manually added users or groups that might or must view iteration nodes.

## **Version control groups and permissions**

By default, the following groups exist at the version-control level:

- ProjectName\**Project Administrators**
- ProjectName\Contributors
- ProjectName\**Readers**
- ProjectName\**Builders**
- TeamProjectCollectionName\Project Collection Administrators
- TeamProjectCollectionName\Project Collection Service Accounts

• TeamProjectCollectionName\**Project Collection Build Service Accounts** 

By default, these groups have the permissions in the following table. For a full description of each permission, see

Permission	By default, set	Consider
Read	Project Administrators, Contributors, Readers, Builders, Project Collection Administrators, Project Collection Service Accounts, Project Collection Build Service Accounts	Any user or group that contributes to the development of this project; any users that should be able to read the contents of a file or folder.
Check Out	Project Administrators, Contributors, Builders, Project Collection Administrators, Project Collection Service	Any user or group that contributes to the development of this project; any users that should be able to

	Accounts, Project Collection Build Service Accounts	check out or make a pending change to items in a folder.
Check In	Project Administrators, Contributors, Builders, Project Collection Administrators, Project Collection Service Accounts, Project Collection Build Service Accounts	Any user or group that contributes to the development of this project; any users that should be able to check in items or revise any committed changeset comments.
Label	Project Administrators, Contributors, Builders, Project Collection Administrators, Project Collection Service Accounts, Project Collection	Any manually added users or groups that might or must label items.

	Build Service Accounts	
Lock	Project Administrators, Contributors, Builders, Project Collection Administrators, Project Collection Service Accounts, Project Collection Build Service Accounts	Any manually added users or groups that might or must lock or unlock folders or files.
Revise other user's changes	Project Administrators, Project Collection Administrators, Project Collection Service Accounts	Manually added users or groups that are responsible for supervising or monitoring the project that might or must change the comments on checked- in files, even if another user checked

		in the file.
Unlock other user's changes	Project Administrators, Project Collection Administrators, Project Collection Service Accounts	Manually added users or groups that supervise or monitor the project and that must be able to unlock files locked by other users.
Undo other user's changes	Project Administrators, Project Collection Administrators, Project Collection Service Accounts	Manually added users or groups that supervise or monitor the project and that must be able to undo a pending change made by another user.
Administer	Project Administrators, Project Collection Administrators,	Manually added users or groups that are supervise or monitor the project and

labels	Project Collection Service Accounts	that must be able to edit or delete labels created by another user.
Manage permissions	Project Administrators, Project Collection Administrators, Project Collection Service Accounts	None.
Check In Other User's Changes	Project Administrators, Project Collection Administrators, Project Collection Service Accounts, Project Collection Build Service Accounts	None.
	Project Administrators, Contributors,	Most users or groups that contribute to

Merge	Builders, Project Collection Administrators, Project Collection Service Accounts, Project Collection Build Service Accounts	the development of this project and that must be able to merge source files, unless the project is under more restrictive development practices.
Manage branch	Project Administrators, Project Collection Administrators, Project Collection Service Accounts, Project Collection Build Service Accounts	Any user or group that contributes to the development of this project and that must be able to create private branches, unless the project is under more restrictive development practices.

## Build-level permissions

By default, the following groups exist at the build level:
- ProjectName\**Project Administrators**
- ProjectName\Contributors
- ProjectName\**Readers**
- ProjectName\**Builders**
- TeamProjectCollectionName\**Project Collection Administrators**
- TeamProjectCollectionName\**Project Collection Build Service Accounts**

By default, these groups have the permissions in the following table. For a full description of each permission, see

Team Foundation Server Permissions.

Permission Name	By default, set for:	Consider adding to:
Administer build permissions	Project Collection Administrators, Project Administrators, Project Collection Build Administrators, Build Administrators,	Any user or group that administers the build servers.
	Project Administrators, Contributors, Readers, Builders, Project	Most manually added users or

View builds	Collection Build Service Accounts, Project Collection Administrators	groups; any that might or must view builds.
Edit build quality	Project Administrators, Contributors, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Retain indefinitely	Project Administrators, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Delete	Project Administrators, Builders, Project Collection	Any user or group that

builds	Build Service Accounts, Project Collection Administrators	administers builds.
Manage build qualities	Project Administrators, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Destroy builds	Project Administrators, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Update build information	Project Collection Build Service Accounts	None.
	Project Administrators,	

Queue build	Contributors, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Manage build queue	Project Administrators, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Stop builds	Project Administrators, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
	Project Administrators, Contributors, Readers,	Most manually added

View build definition	Builders, Project Collection Build Service Accounts, Project Collection Administrators	users or groups; any that might or must view build definitions.
Edit build definition	Project Administrators, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Delete build definition	Project Administrators, Builders, Project Collection Build Service Accounts, Project Collection Administrators	Any user or group that administers builds.
Override check-in validation	Project Collection Build Service Accounts, Project	Any user or group that administers

by build Collection builds. Administrators

#### **Lab Management groups and permissions**

By default, the following groups exist at the lab management level:

- ProjectName\**Project Administrators**
- ProjectName\**Contributors**
- ProjectName\**Readers**
- TeamProjectCollectionName\**Project Collection Administrators**
- TeamProjectCollectionName\Project Collection Build Service accounts
- Server\Team Foundation Administrators

By default, these groups have the permissions in the following table. In addition, the creator of an object in Lab Management is automatically granted all permissions on that object. For a full description of each permission, see

Team Foundation Server Permissions.

Permission	By default, set	Consider
Name	for:	adding to:
View Lab Resources	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors,	All manually- added users or groups that need to view Lab resources

	Readers, Project Collection Build Service accounts	
Manage Lab Locations	Team Foundation Administrators, Project Collection Administrators, Project Administrators (limited to only project-level locations, that is, project host group and project library share)	Most manually added users or groups; any that might or must administer Lab assets
Delete Lab Locations	Team Foundation Administrators, Project Collection Administrators Project Administrators (limited to project-level locations such as project host groups and project library	Most manually added users or groups; any that might or must administer Lab assets

	shares)	
Write Environment and Virtual Machine	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors, Project Collection Build Service accounts	Most manually added users or groups; any that might or must operate on Lab environments
Edit Environment and Virtual Machine	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors, Project Collection Build Service accounts	Most manually added users or groups; any that might or must operate on Lab environments
Delete Environment and Virtual	Team Foundation Administrators, Project Collection Administrators,	Most manually added users or groups; any that might or

Machine	Project Administrators	must operate on Lab environments
Import Virtual Machine	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors	Most manually added users or groups; any that might or must operate on Lab environments
Environment Operations	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors, Project Collection Build Service accounts	Most manually added users or groups; any that might or must operate on Lab environments
Manage Permissions	Team Foundation Administrators, Project Collection Administrators	Administrators who must administer Lab assets

Manage Child Permissions	Team Foundation Administrators, Project Collection Administrators, Project Administrators (limited to only project level locations, that is, project host group and project library share)	Administrators who must administer Lab assets
Start	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors, Project Collection Build Service accounts	Most manually added users or groups; any that might or must operate on Lab environments
Stop	Team Foundation Administrators, Project Collection Administrators, Project	Most manually added users or groups; any

	Administrators, Contributors, Project Collection Build Service accounts	that might or must operate on Lab environments
Pause	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors, Project Collection Build Service accounts	Most manually added users or groups; any that might or must operate on Lab environments
Manage snapshots	Team Foundation Administrators, Project Collection Administrators, Project Administrators, Contributors, Project Collection Build Service accounts	Most manually added users or groups; any that might or must operate on Lab environments

## See Also

#### Tasks

Set Administrator Permissions for Team Foundation Server Add Users to Team Projects Change Permissions for a Group or User View Permissions

#### Concepts

Team Foundation Server Permissions

#### **Other Resources**

<u>Changing Groups and Permissions with TFSSecurity</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Change Permissions for a Group or User
See Also Send Feedback

You can explicitly change permissions for a user or group at the project level, the collection level, or the server level in Visual Studio Team Foundation Server. If you change the permission for a group, you will potentially affect the permissions for all of the users who are members of that group. For example, a user might be a member of two groups in a project. If one group has a permission set to Deny and the other group has the same permission set to Allow, the user will be denied that permission. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

If you change permissions for users and groups in Team Foundation Server, you do not automatically change any permissions that they might have in SQL Server Reporting Services or SharePoint Products. You must manually change permissions for users and groups in those products.

#### **Required permissions**

To perform this procedure, you must be a member of the administrative group at the level of the group that you want to change. For example, if you want to change permissions for a group or user at the collection level, you must be a member of the Project Collection Administrators group for that collection, or your Edit Collection-Level Information permission must be set to Allow. For more information about permissions, see <u>Team Foundation Server Permissions</u>.

#### To change project-level permissions for a group or user

1. In Team Explorer, open the team project for which you want to change permissions for a group or user, and then choose Settings.

The Settings menu appears.

2. In the Settings menu, in Team Project, choose Security.

Team Web Access opens in the administration context for the team project, on the security tab.

3. On the groups tab, choose the group whose permissions you want to modify. Alternately, on the users tab, choose the user whose permissions you want to modify.

Once you have chosen a group or user, on the permissions tab, choose the permission you want to modify, and then toggle the value of that permission to the value you want to set for that user or group.

You can choose between the following values:

- 1. allow
- 2. deny
- 3. not set
- 4. inherited allow or inherited deny, if either of these values was the original value for the permission

#### ĭTip

If the value for a permission is inherited allow or inherited deny, you can choose Why? to trace where the inheritance originates.

4. When you have finished modifying permissions, choose Save Changes.

#### To change collection-level permissions for a group or user

1. In Team Explorer, choose Team, choose Team Project Collection, and then

choose Security.

Team Web Access opens in the administration context for the team project collection, on the security tab.

2. On the groups tab, choose the group whose permissions you want to modify. Alternately, on the users tab, choose the user whose permissions you want to modify.

Once you have chosen a group or user, on the permissions tab, choose the permission you want to modify, and then toggle the value of that permission to the value you want to set for that user or group.

You can choose between the following values:

- 1. allow
- 2. deny
- 3. not set
- 4. inherited allow or inherited deny, if either of these values was the original value for the permission

#### ĭTip

If the value for a permission is inherited allow or inherited deny, you can choose Why? to trace where the inheritance originates.

3. When you have finished modifying permissions, choose Save Changes.

#### To modify permissions for groups at the server level

- 1. Open the administration console for Team Foundation on the applicationtier server where you want to modify permissions for a server-level group.
- 2. Choose Application Tier, and then choose Administer Security.

The Global Security window opens.

3. Under Users and Groups, choose the group for which you want to modify permissions.

#### ⊠Note

If the group that you want to modify does not appear in the list, you must add it by using the controls in the Add users and groups section.

4. In Permissions, select or clear the Allow or Deny check box for the permission that you want to modify.

#### ĭNote

If neither the Allow check box nor the Deny check box is selected, the permission level is set to Not Set. For more information, see <u>Team</u> <u>Foundation Server Permissions</u>.

5. When you finish modifying permissions, choose Close.

## See Also

#### Concepts

Configuring Users, Groups, and Permissions

#### **Other Resources**

How to: Add Users to Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
View Permissions
See Also Send Feedback

When administering Team Foundation Server, you might want to view the permissions granted to individual users or groups. This information can help you better understand the levels of access users and groups have to the deployment, and help you troubleshoot access problems. You can view permissions in one of two ways: through Team Web Access, or by using the administration console for Team Foundation Server. You can access security and permissions for the team, project, and collection levels through either Team Explorer or Team Web Access. For permissions at the team level, you must use Team Web Access. For permissions at the server level, you must use the administration console.

#### ĭTip

In addition to permissions, your licensing level can determine what features are available for your users. For more information about viewing licensing levels, see <u>Add Users to Team Projects</u>.

#### **Required Permissions**

To perform this procedure, you must be a member of the Project Collection Valid Users group or have the View Project-Level Information permission set to Allow. For more information about permissions, see <u>Team Foundation Server</u> <u>Permissions</u>.

#### To view collection-level permissions

- 1. Do one of the following:
  - In Team Explorer, under Team Project Collection, choose Security.

A browser window opens the administration menu for Team Web Access.

- In Team Web Access, open the administration menu, and in the context menu, choose the name of the team project collection.
- 2. In the hub menu area, choose the Security tab.
- 3. In Members, choose groups or users, and then choose the user or group for which you want to view permissions.

Permissions appear in the Summary pane. If you want to view the members of a group, choose the members tab. If you want to view the groups to which that user or group belongs, choose the member of tab.

The following illustration shows the summary view of the Project Collection Administrators group for the fictitious company, Fabrikam Fiber:



#### To view project-level permissions

- 1. Do one of the following:
  - In Team Explorer, under Team Project, choose Security.

A browser window opens the administration menu for Team Web Access.

- In Team Web Access, open the administration menu, and in the context menu, choose the name of the team project.
- 2. In the hub menu area, choose the Security tab.
- 3. In Members, choose groups or users, and then choose the user or group for which you want to view permissions.

Permissions appear in the Summary pane. If you want to view the members of a group, choose the members tab. If you want to view the groups to which that user or group belongs, choose the member of tab.

#### To view team-level permissions

- 1. Open Team Web Access, connect to your project, and in the context menu, choose the administration icon, and then choose Settings.
- 2. In the hub menu area, choose the Overview tab.
- 3. In Overview, review the list of administrators for the team, and the list of team members.

No team permissions are explicitly visible. The only difference between permissions for a team administrator and for team members is that team administrators are the only users who can add or remove members from the team. If you choose the Security tab, you will see the permissions for each team member at the project level, not the team level.

#### To view server-level permissions

- 1. Open the administration console and choose Application Tier.
- 2. In the Application Tier pane, choose Administer Security.
- 3. In Users and Groups, choose the user or group for which you want to view permissions.

Permissions will appear in the Permissions area.

## **See Also**

#### Tasks

Change Permissions for a Group or User

#### **Other Resources**

Managing Permissions

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 Visual Studio Application Lifecycle Management
 Configuring and Managing Your Build System
 Send Feedback

Do you want to automatically and consistently build, test, and deploy your app in a distributed environment? To get your team started using Team Foundation Build, you need one or more build servers in your build system.



#### ĭTip

If your team project collection is hosted on <u>Team Foundation Service</u> you might be able to use the Hosted Build Controller instead of deploying your own build system.

## **What do you want to do?**

Deploy and work with a build server

To use Team Foundation Build with an on-premises Team Foundation Server, you must deploy and dedicate at least one build server to your team project collection.

Deploy and work with a build controller

Use a build controller to perform lightweight tasks and distribute the processor-intensive work of your build process to its pool of build agents.

Deploy and work with build agents

Use build agents to do the processor-intensive work of your build, including provisioning the workspace, getting files from version control, compiling the code, and running tests.

Set up drop folders

Prepare and then designate one or more drop folders so that your build system can deliver binaries and log files to your team.

Scale out your build system

As your team and your code base grow, expand your build system incrementally with relative ease.

#### Manage your build system

Occasionally, you will need to monitor and manage your build system, confirm the health of your build server, or diagnose problems.

Use your build system to compile, test, and deploy your app

After your build system is in place, use it to automatically compile and test your applications, and perform other important functions.

**Receive Build Notifications** 

Sign up for email alerts for yourself, and if you have sufficient permissions, for your team.

Start, stop, delete, or destroy builds from the command prompt

Use the command prompt directly or use a script. (We are not currently republishing this guidance. However, you can read the Visual Studio 2010 versions of the topics.)

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Deploy and Configure a Build Server
Send Feedback

To use Team Foundation Build with an on-premises Team Foundation Server, you must deploy at least one build server.

#### ĭTip

If your team project collection is hosted on <u>Team Foundation Service</u> and your team's needs can be met by a single standard build agent, you can use the Hosted Build Controller instead of deploying your own build agent.

Each build server serves a single team project collection. In fact, although you configure, modify, and manage a build server directly on the computer where Team Foundation Build Service is running, the configuration data is stored in the team project collection.



On a build server, you can run:

- 1. A single build controller
- 2. One or more build agents
- 3. A single build controller and one or more build agents

You can host a build server on the same computer as your Team Foundation Application-Tier Server, but, in most of these situations, this build server should not host any build agents. Build agents place heavy demands on the processor, which could significantly decrease the performance of your application tier. In addition, you might want to avoid running build server components on the application tier to avoid <u>increasing the attack surface</u>. For more detailed examples of viable build system topologies, see <u>Scale out Your Build System</u>.

#### **Required Permissions**

You must be a member of the Windows Administrators group on the build server and a member of the Project Collection Build Administrators group on your team project collection. See <u>Team Foundation Server Permissions</u>. What do you want to do?

- <u>Understand security risks</u>
- <u>Deploy a build server</u>
- <u>Begin configuring a build server</u>
- Connect a build server to a team project collection
- <u>Specify service accounts</u>
- <u>Run your build server in interactive mode</u>
- <u>Take next steps</u>

# **Understand security risks**

Installing Team Foundation Build Service increases the attack surface of the computer. Because developers are treated as trusted entities in the build system, a malicious user could, for example, construct a build definition to run arbitrary code that is designed to take control of the server and steal data from Team Foundation Server. Customers are encouraged to follow security best practices as well as deploy defense in-depth measures to ensure that their build environment is secure. This includes developer workstations. For more information regarding security best practices, see the TechNet Article

Security Guidance.

## **Deploy a build server**

You deploy a build server by installing the Team Foundation Build Service. Before you begin this process, here are some tips.

Have you installed Visual Studio 2012 Update 2 on your on-premises application-tier server?

- Yes
  - You can connect a Team Foundation Build 2010 server to your onpremises Team Foundation 2012 application-tier server.
  - You cannot run both Team Foundation Build 2010 and Team Foundation Build 2012 on the same computer.
- No
  - Only a build server that is running Team Foundation Build 2012 can connect to Team Foundation Server 2012 application tier.
  - Although a build server that is running Team Foundation Build 2010 cannot connect to your application tier, you can run both Team Foundation Build 2010 and Team Foundation Build 2012 on the same computer.

#### ⊠Note

Upgrade template builds might not function correctly in this kind of side-by-side configuration.

#### Some more tips:

• If you install the build service while you are logged on as a member of the Project Collection Administrators, the installation automatically adds the build service account to the Project Collection Build Service Accounts

group, so you don't need to do it manually.

- You can replace an existing build server by copying its configuration to the new build server. See Install Team Foundation Build Service.
- You can set up an ad-hoc build server on any client or server computer that has adequate processing and storage capacity. For example, an individual developer who has an extra computer could set it up as a build server.
- You can deploy a build server on a physical computer or a virtual machine.

For step-by-step instructions to deploy a build server, see Install Team Foundation Build Service.

# Begin configuring a build server

After you deploy your build server, you can configure it to meet your team's needs.

- 1. Log on to the build server that you want to configure.
- 2. From Windows Start, run Team Foundation Administration Console.

The Team Foundation Administration Console appears.

- 3. In the tree pane, expand the name of the server.
- 4. Choose the Build Configuration node.



#### ⊠Note

If the message () Configure Installed Features appears instead of a build

controller or build agents, as shown above, see

#### Deploy a build server.

• Choose Properties.



The Build Service Properties dialog box appears.

uild Service Properties		? <b>×</b>
Build Service on fabrikam-1 is Running Stop the service to make changes		
Communications:		
Provide Build Services for Project Collection	n:	
http://fabrikam-3:8080/tfs/defaultcollection	n Brows	e
Listen for Build Agent Communication on:		
http://fabrikam-1.northamerica.corp.micros	oft.com Chang	je
NT AUTHORITY\NETWORK SERVICE	Chang	je
Connect to Team Foundation Server as:		
Vise the same identity as Windows Servi	ice	
	Chang	je
Run the Service Interactively - when is	this useful?	
	Start CI	ose

Before you can configure the build server, you must choose the Stop the service link. See the sections below for details about how to configure your build server.

# ■Connect a build server to a team project collection

Under Communications, next to Provide Build Services for Project Collection, choose the Browse button to connect your build server to a team project collection on an on-premises Team Foundation Server or on

Team Foundation Service.

You can strengthen security by using Hypertext Transfer Protocol Secure (HTTPS) with Secure Sockets Layer (SSL). See <u>Setting up HTTPS with Secure</u> <u>Sockets Layer (SSL) for Team Foundation Server</u>.

# **Specify service accounts**

Under Run the Service as you can specify the accounts that enable the build server to provide its services.

#### **Specify the build service account**

Immediately under Run the Service as, you can specify the build service account.

#### NETWORK SERVICE account

For most purposes, the best setting is NT AUTHORITY\NETWORK SERVICE.

un the Service as:	
NT AUTHORITY\NETWORK SERVICE	Change

One advantage of this approach is that if someone changes the password of a user account (some network administrators require such a change on a regular basis), the build server does not go offline.

User account

Occasionally, you might be required to specify a user account, such as NORTHAMERICA\FABBUILD.

	Run the Service as:
Change	DOMAINB\FABBUILD
	DOMAINB/FABBUILD

Examples of situations where you must specify a user account include:

• You want to run your build server in interactive mode,

as explained below.

• Your Team foundation Server is inside your firewall, but the build server is outside your firewall.

Regardless of the account you specify, the build service account must belong to the Project Collection Build Service Accounts group.

#### **Specify the account used to connect to your Team Foundation** Server

You can usually leave the second text box empty. However, in the following cases, your build server can't connect to your Team Foundation Server using the build service account.

Domain trust differences: The domain of the Team Foundation Server does not trust the domain of the build server. For example, the build server is in domainb, and Team Foundation Server is in domaina, which does not trust domainb. You could specify the build service account in the first box, and an account from domaina in the second box:

domainb\fabbuild	Change
Connect to Team Equindation Converses	
Connect to Team Foundation Server as:	
Connect to Team Foundation Server as: Use the same identity as Windows Service	

Team project collection hosted on Team Foundation Service: When you connect your on-premises build server to

<u>Team Foundation Service</u>, then the Use same identity as Windows Service check box is automatically cleared and the account you used to connect to Team Foundation Service (for example, a Windows Live account) is specified beneath it.

# ■Run your build server in interactive mode

For most purposes, you should run your build server as a Windows service, which is the default setting. However, there are a few tasks that a build agent can perform only on a build server that is running as an interactive process.

#### To run your build server in interactive mode

- 1. Identify the user account that will act as the build service account. The build service account must:
  - Be a member of the Windows Administrators group on the build server.
  - Be a member of the Build Service Accounts group on your team project collection. See <u>Grant a Build Server Permission to Serve a</u> <u>Team Project Collection</u>.
  - Have Change and Read privileges on the drop folder, if any, that you plan to specify in your build definition. See <u>Select a Staging Location</u> and <u>Set Up a Drop Folder</u>.
- 2. On the Build Service Properties dialog box, choose Stop the service.
- 3. Under Run the Service as, choose Change, and then specify the credentials of the build service account.
- 4. Select Run the Service interactively.


- 5. Choose Start, and then choose OK.
- 6. Leave the build service account logged on to the build server.

## **Next Steps**

Deploy and configure a build controller

Use a build controller to perform lightweight tasks and distribute the processor-intensive work of your build process to its pool of build agents. You can host one build controller on a build server.

Deploy and configure build agents

Use a build agent to do the processor-intensive work of your build, includes getting files from version control, provisioning the workspace, compiling the code, and running tests. You host can one or more build agents on a build server.

Set up drop folders

You can prepare and then designate one or more drop folders so that your build system can deliver binaries, test results, and log files to your team.

Scale out your Team Foundation Build system

As your team and your code base grow, you can expand your build system incrementally, with relative ease.

#### Manage your build system

After you deploy your build server, you can manage it from the Team Foundation Administration Console. You can manage the build controller and build agents from either Team Foundation Administration Console or from Visual Studio.

## Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Grant a Build Server Permission to Serve a Team Project Collection
Send Feedback

#### ĭTip

You can skip this topic if you deploy your build server while you are logged on as a member of the Project Collection Administrators group.

A build server can build and test code in a team project collection only if the build service account is a member of the Project Collection Build Service Accounts group of that collection.

**Required** permissions

You must be a member of the Project Collection Administrators group. See <u>Team Foundation Server Permissions</u>.

# ■Grant a build server permission to serve a team project collection

- 1. In Visual Studio, in Team Explorer:
  - 1. If you are not already connected to a team project in the team project collection, then connect to the team project.
  - 2. Choose 🏠 Home, and then choose 🍄 Settings.
- 2. Under Team Project Collection, choose Group Membership.

Your web browser displays the Security tab and the Groups sub-tab.

3. Under TFS Groups, choose Project Collection Build Service Accounts.

The members of the group appear in the pane on the right side. Verify that your build service account is already present. If not, continue to the next step.

- 4. Choose Add, and then choose Add windows user or group.
- 5. On the Add a windows User or Group dialog box, type one of the following values into the Identities box, depending on the type of build service account that your build server uses:
  - User account: Type the account name into the box. For example: NORTHAMERICA\FABBUILD
  - NETWORK SERVICE on a computer other than your Team Foundation Server: Type computer\_name\$. For example, if your computer is named FABRIKAM-1, then you would specify: FABRIKAM-1\$
  - NETWORK SERVICE on the same computer as your Team Foundation Server: Type NETWORK SERVICE.

6. Choose Check Name to confirm that the name is valid, and if it is valid, choose Save Changes.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Deploy and Configure a Build Controller
Send Feedback

To use Team Foundation Build, your team must have at least one build controller to perform lightweight tasks and distribute the processor-intensive work of your build process to its pool of build agents.



Each build controller is dedicated to a single team project collection. The build controller performs some lightweight tasks, such as determining the name of the build, creating the label in version control, logging notes, and reporting status from the build. The build controller uses the AgentScope activity to delegate processor-intensive work, such as compiling code or running tests, to the build agents within its pool. These build agents can be hosted either on the same build server, or on a different build server.

Because a build controller does not typically require significant processor time, in many cases you can host it on the same computer as your Team Foundation Server, as shown above, or on a low-powered physical or virtual machine.

However, a build controller can demand a significant amount of memory in

certain situations, so you should provide sufficient memory as needed to ensure it functions properly. Or, depending on your team's needs and resources, you might want to run the build controller on a different computer than your Team Foundation Server.



#### **Required Permissions**

You must be a member of the Windows Administrators group on the build server, and a member of the Project Collection Build Administrators group on your team project collection. See <u>Team Foundation Server Permissions</u>.

What do you want to do?

- Create or modify a build controller
- Enable your build processes to leverage supplemental binaries
- Specify the maximum number of concurrently running builds
- <u>Remove a Build Controller</u>
- <u>Use a build controller with a team project collection hosted on Team</u> <u>Foundation Service</u>

# **Create or modify a build controller**

### To create or modify a build controller from the build server

- 1. Log on to the build server.
- 2. From Windows Start, run Team Foundation Administration Console.
- 3. In the tree pane of the Team Foundation Administration Console, expand the name of the server, and then choose the Build Configuration node.

⊠Note

If the **()** Configure Installed Features message appears, see Deploy a build server.

- 4. On the Build Configuration page:
  - If a controller is not listed, choose New Controller.
  - If a controller is already listed:



Choose Properties.

The Build Controller Properties dialog box appears.

## To modify a build controller from Visual Studio

- 1. In Visual Studio, in Team Explorer:
  - 1. If you are not already connected to a team project in the team project collection, then connect to the team project.

- 2. Choose 🖆 Home, and then choose 🚔 Builds.
- 2. On the Builds page, choose Actions, and then Manage Build Controllers.

The Manage Build Controllers dialog box appears.

Name 🔺	Status	State	Date updated	Properties
Default Controller - FABRIKAM-1	Available	Enabled	7/12/2012 2:35 PM	Remove
Default Agent - FABRIKAM-1	Available	Enabled	7/12/2012 2:36 PM	
Default Agent1 - FABRIKAM-1	Available	Enabled	7/12/2012 3:36 PM	Refresh
FABRIKAM-3 - Agent1	Available	Enabled	7/12/2012 3:37 PM	
Default Controller - FABRIKAM-3	Available	Enabled	7/12/2012 9:33 AM	
Default Agent - fabrikam-8rp	Available	Enabled	7/12/2012 1:56 PM	
escription:				
aas				Close

3. Select the build controller that you want to modify, and then choose Properties.

The Build Controller Properties dialog box appears.

uild Controllers Properties	? 💌
Display Name:	
Default Controller - FABRIKAM-1	
Description:	
Connector Name	
Computer Name:	
FABRIKAM-1	
Version control path to custom assemblies:	
Maximum number or concurrently running builds     O Default to number of enabled agents     Specify the maximum	
Build Controller service is enabled Build Controller status is Offline	
ОК	Cancel

In the Display Name and Description fields: Type a name and a description that can help team members easily identify the appropriate build controller when they define their build processes.

See the sections below for details about how to configure your build controller.

# **Enable your build processes to leverage supplemental binaries**

You can more deeply customize your build process by leveraging:

- Assemblies that contain your custom workflow activities
- Custom MSBuild tasks
- Third-party unit test frameworks in your test runs.

To enable your build processes to leverage these kinds of code, check the binaries in to the folder (or any of its descendant folders) that you specify in the Version control path to custom assemblies box. When you set or modify the value in this box, the build server automatically restarts to load the assemblies.

#### ĭTip

If your build results include a TF215097 error message, your build process might include a custom activity that is not in this folder or in any of its descendant folders.

For guidance about how to check in the binaries to version control, see Leverage third-party binaries that your code does not build.

# ■Specify the maximum number of concurrently running builds

In most cases, you should leave this value set to Default to number of enabled agents. However, in some cases, you might want to control the depth of the build queue. For example:

- You observe that too many builds show a status of Running even though many of them are stalled as they wait for a build agent to be assigned to them. This situation can confuse the team. To avoid the problem, select Specify the maximum and then specify a value that is lower than the number of build agents that are pooled under the build controller.
- You design a custom build template to enable your builds to leverage more than one build agent at a time, from a pool of 12 build agents under the build controller. The builds that are processed by this build controller are based on a custom build template that concurrently delegates to three build agents. To make the build queue data more useful, you could configure the build controller with Maximum number of concurrently running builds set to Specify the maximum, with a value of 4.

## **Remove a Build Controller**

1. In Visual Studio, open the Manage Build Controllers dialog box, as explained above in

Modify a build controller from Visual Studio.

- Select and then remove each build agent under the build controller, either by choosing Remove, or by choosing Properties and then using the Build Agent Properties dialog box to assign the build agent to another build controller.
- Select the build controller that you want to remove, and then choose Remove.

#### ⊠Note

You can also use the Team Foundation Administration Console to remove the build controller while you are logged on to the build server. But if you do, you might still need to use Visual Studio, or log on to other build servers first, to remove any remote build agents pooled by the build controller.

# ■Use a build controller with a team project collection hosted on Team Foundation Service

Use the Hosted Build Controller: If your team project collection is hosted on

<u>Team Foundation Service</u>, you might be able to skip deploying your own build controller and use the Hosted Build Controller instead. To use the Hosted Build Controller, you simply select it when you define your build process. See <u>Use the Hosted Build Controller</u>.

Use an on-premises build controller: If your build process requires resources outside of the <u>Hosted Build Controller capabilities</u>, then you can connect your on-premises build controller to your hosted team project collection.



## **Next Steps**

Deploy and configure build agents

Use a build agent to do the processor-intensive work of your build. This includes getting files from version control, provisioning the workspace, compiling the code, and running tests. You can host one or more build agents on a build server.

Set up drop folders

You can prepare and then designate one or more drop folders so that your build system can deliver binaries, test results, and log files to your team.

Scale out your Team Foundation Build system

As your team and your code base grow, you can expand your build system incrementally, with relative ease.

Manage your build system

After you deploy your build server, you can manage it from the Team Foundation Administration Console. You can manage the build controller and build agents from either Team Foundation Administration Console or from Visual Studio.

#### Define your build process

After your build system is in place, you are ready to define your build process, which contains your instructions about which code projects to compile, what actions trigger a build, what tests to run, and any other procedures required by your team.

#### Send

feedback on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Deploy and Configure Build Agents
Send Feedback

To use Team Foundation Build, your team must have at least one build agent to perform the processor-intensive work of your build process.



Each build agent is dedicated to and controlled by a single build controller. Build agents can be hosted on the same build server that hosts their build controller, but this is not required, and in some cases your team's needs can most efficiently be met by a single build server to host a build controller that controls build agents on multiple build servers.

The build agent executes the steps of your build process that are contained in the AgentScope activity. Typically, these steps include getting files from version control, provisioning the workspace, compiling the code, running tests, and merging files back into version control.

Make sure the build server that hosts your build agents has sufficient storage and processing capabilities to match the size and complexity of the codebase and tests on the team project collection. Typically, you should host no more than one build agent per processor core on the build server. You can also enhance performance by dedicating a single physical hard drive to the <u>working directory</u> of each build agent.

### ĭTip

If your team project collection is hosted on <u>Team Foundation Service</u>, and your team's needs can be met by a single standard build agent, you can use the Hosted Build Controller instead of deploying your own build agent.

#### **Required permissions**

You must be a member of the Windows Administrators group on the build server and a member of the Project Collection Build Administrators group on your team project collection. See <u>Team Foundation Server Permissions</u>.

What do you want to do?

- Create or modify a build agent
- Install Visual Studio and other software to enable compilation and other capabilities
- <u>Specify the working directory</u>
- Enable your build agent to run tests
- <u>Assign tags to represent build agent capabilities or purposes</u>
- Deploy a build agent that can compile and test a Windows Store app
- <u>Remove a build agent</u>

## **Create or modify a build agent**

### To create or modify a build agent from the build server

- 1. Log on to the build server that you want to configure.
- 2. From Windows Start, run Team Foundation Administration Console.

The Team Foundation Administration Console appears.

3. In the tree pane of the Team Foundation Administration Console, expand the name of the server, and then choose the Build Configuration node.

Information about the build server appears in the content pane.

If the message () Configure Installed Features appears, see Deploy a build server.

- 4. On the Build Configuration page:
  - To create a new build agent, choose New Agent.
  - To modify an existing build agent



choose Properties.

The Build Agent Properties dialog box appears.

## To modify a build agent from Visual Studio

- 1. In Visual Studio, in Team Explorer:
  - 1. If you are not already connected to a team project in the team project collection, then connect to the team project.

- 2. Choose 🖆 Home, and then choose 🚔 Builds.
- 2. On the Builds page, choose Actions, and then Manage Build Controllers.

The Manage Build Controllers dialog box appears.

Name 🔺	Status	State	Date updated	Properties
Default Controller - FABRIKAM-1	Available	Enabled	7/12/2012 2:35 PM	Remove
Default Agent - FABRIKAM-1	Available	Enabled	7/12/2012 2:36 PM	
Default Agent1 - FABRIKAM-1	Available	Enabled	7/12/2012 3:36 PM	Refresh
FABRIKAM-3 - Agent1	Available	Enabled	7/12/2012 3:37 PM	
Default Controller - FABRIKAM-3	Available	Enabled	7/12/2012 9:33 AM	
Default Agent - fabrikam-8rp	Available	Enabled	7/12/2012 1:56 PM	
escription:				

3. Select the build agent that you want to modify, and then choose Properties.

The Build Agent Properties dialog box appears.

Build Agent Properties	? ×
Display Name: Default Agent - FABRIKAM-1 Description:	Tags: Add new tag
Controller:	
Default Controller - FABRIKAM-1	]
Computer Name:	
FABRIKAM-1	
Working Directory:	
\$(System Drive)\Builds\\$(BuildAgentId)\\$(BuildDefinit	
Build Agent service is enabled	
Build Agent status is Available	1
DDUESRV Team Foundation Service Account changed the agent status.	
	OK Cancel

Display Name, Description: Type a name and a description to help team members easily identify the build agent.

Controller: Select the build controller that you want to control this build agent. The build controller can run on the same build server as this build agent or on a different build server.

See the sections below for details about how to configure your build agent.

# Install Visual Studio and other software to enable compilation and other capabilities

It is a best practice to install on the build agent the version of Visual Studio that your team uses on its dev machines. See Installing Visual Studio. You should also install any other software and components that are installed on your dev machines and that are required to build your app. Otherwise, you might have problems, such as some of your code projects failing to compile.

# **Specify the working directory**

You can specify the working directory, which the build agent uses to read from or write to files. For example, source files are copied to subdirectories in this folder, and binaries are created and stored in other subdirectories in this folder.

#### ĭTip

You can enhance performance by dedicating a single physical hard drive to the

working directory of each build agent.

#### **Use working directory tokens**

Although you can specify a literal path for the Working Directory property (for example, c:\temp\build\), a simpler and more durable approach is to use tokens to specify the path. You can use two kinds of tokens:

Environment variables

Environment variables contain information about the environment for the system and the user who is logged on. The most typical variable you might use is **SYSTEMDRIVE**, but for some situations you might also use variables such as **USERNAME** or **HOMEPATH**.

#### ĭTip

To list the environment variables on your build server, open a command prompt and type **set**.

Team Foundation Build variables

You can use the following variables in a build agent working directory:

• **\$(BuildAgentId)**: An automatically generated integer that uniquely identifies a build agent within a team project collection.

- **\$(BuildAgentName)**: The Display Name of the build agent.
- **\$(BuildDefinitionId)**: An automatically generated integer that uniquely identifies a build definition within a team project collection.
- **\$(BuildDefinitionPath)**: The team project name and the build definition name, separated by a backslash.

## **Working directory example**

For example, you have a build agent that is named BuildBot3. You have defined two builds that are called NightlyBuild and WeeklyBuild in a team project that is called CoolApp. In the Working Directory box, you specify the following value: \$(SystemDrive)\TeamBuilds\\$(BuildAgentName)\\$(BuildDefinitionPath). As a result, the BuildBot3 build agent creates and uses the following two working directories:

#### C:\TeamBuilds\BuildBot3\CoolApp\NightlyBuild

#### C:\ TeamBuilds\BuildBot3\CoolApp\WeeklyBuild

## **■Make sure the path to the working directory is not too long**

The working directory that you specify must not cause the build agent to produce any physical paths that are longer than 259 characters. Otherwise, your builds could fail and log this error: **TF10128: The path** PhysicalPath **contains more than the allowed 259 characters. Type or select a shorter path**.

To solve this problem, specify a working directory that results in a shorter physical path. For example you could specify \$(HOMEDRIVE)\bld\\$(BuildAgentID)\\$(BuildDefinitionID), which would result in a working directory such as: c:\bld\3\2\.

## **Subdirectories created in the working directory**

The build agent creates and works in the following subdirectories under this path.

Subdirecto	ory Used to Store Files
Sources	Read by the build agent, such as source files. You specify the files that it downloads in the Workspace settings for each build definition. See Work with Build Workspaces.
Binaries	Compiled by the build agent, such as compiled application files.
TestResult	s Produced by whatever tests the build agent ran.

# **Enable your build agent to run tests**

You can define a build process that performs one or more automated test runs.

#### **▲Important**

Many kinds of tests and test operations require that you install on your build agent the same version of Visual Studio that your team uses on its dev machines. See Installing Visual Studio.

The build agent can run:

- Code Coverage
- Coded UI Tests (Requires a build server that is running in interactive mode. See Run your build server in interactive mode and Verifying Code by Using Coded User Interface Tests.)
- Database Test Data Generation
- Database Unit Tests
- Generic Tests
- Load Tests
- Unit Tests
- Ordered Tests
- Test Impact Analysis
- Web Tests

# ■Assign tags to represent build agent capabilities or purposes

As the scale of your build system grows, you might need to define specialized build agents. Whenever a build agent has special capabilities or is intended for a specific kind of usage, you should assign one or more tags to that agent. This way, when a team member creates a build definition that requires a certain kind of build agent, they can specify the tag in their build definition.

You can assign tags from the

I

<u>Build Agent properties dialog box</u>, described above. You can then apply the tags to your build definitions.

The following table offers examples of tag names and the build agent capabilities that they could represent.

You could apply the following tag 	To identify a build agent that can
x86	Compile 32-bit applications
x64	Compile 64-bit applications
bvt	Run your BVT tests, which are run by your nightly BVT build.

WindowsStore	Compile and test a Windows Store app.
IIS	Compile an ASP.NET Web application and then stage and host it on the computer on which the build agent is running.
interactive	Perform tasks that require an agent on a build server that is running in interactive mode.

You can apply more than one tag to a build agent. For example, you could create a build agent with the tags x86 and Release to indicate an agent that is set up to compile the Release configuration of a 32-bit application.

If you run multiple build agents on the same build server, they probably will all have the same capabilities. Therefore, you probably will want to apply the same tags to all build agents on that build server.

## **Remove a build agent**

1. In Visual Studio, open the Manage Build Controllers dialog box, as explained previously in

Create or modify a build agent.

• Select the build agent that you want to remove, and then choose Remove.

#### ĭTip

You can also use the Team Foundation Administration Console to remove the build agent while you are logged on to the build server.

## **Next steps**

Scale out your Team Foundation Build system

As your team and your code base grow, you can expand your build system incrementally, with relative ease.

Manage your build system

Occasionally, you will need to monitor and manage your build system.

Use your build system to compile, test, and deploy your app

After your build system is in place, your team is ready to create a simple build process (for example, a continuous integration build) and benefit from automated building and testing of your app.

Send

feedback on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Select a Staging Location and Set Up a Drop Folder
Send Feedback

When you define or queue a build definition, you can specify a path to a drop folder so that your build process can deliver binaries and log files to your team. Make sure the folder you've identified is prepared for use as a drop folder.

What do you want to do?

- Set up a drop folder on a file share for your on-premises build controllers
- Drop outputs into TFS when you use the Hosted Build Controller

# Set up a drop folder on a file share for your on-premises build controllers

You can specify a path to a file share when you define or queue a build definition that is run by an on premises-build controller. The on-premises build controller can be dedicated to a team project collection on either an on-premises Team Foundation Server (TFS) or on Team Foundation Service.



#### ĭTip

If you see a TF270016 error in your build results log, it's possible that the build server does not have the required permissions to access the drop folder. Follow the instructions below to resolve this problem.

## **Requirements**

You must be a member of the Administrators group on the computer where the drop folder resides.

You can use a file share drop folder on the following operating systems:

- Windows Server 2012
- Windows Server 2008
- Windows Server 2003
- Windows 8
- Windows 7
- Windows Vista

## To set up a drop folder on a file share

- 1. Log on to the computer where the drop folder will reside.
- 2. Open Windows Explorer (File Explorer in Windows 8), and browse to the folder that contains (or will contain) the drop folder.
- 3. If necessary, create the drop folder.

### ĭTip

Try to keep the path to the drop folder as short as possible (for example: c:\drops). The drop folder must not cause the build agent to produce any physical paths that contain more than 259 characters. Otherwise, your builds will fail and log the **TF205022** error message.

- 4. Open the shortcut menu for the folder, choose Properties, and then choose the Sharing tab.
- 5. On Windows Server 2012, Windows Server 2008, Windows 8, Windows 7,

and Windows Vista only: Choose Advanced Sharing.

The Advanced Sharing dialog box appears.

6. Select Share this folder, and then choose Permissions.

The Permissions for FolderName dialog box appears.

7. Choose Add.

The Select Users, Computers, or Groups dialog box appears.

- 8. Perform one of the following steps, depending on what type of account you've specified as the build service account on the build server that hosts your build controller:
  - NETWORK SERVICE running on the computer that contains the drop folder: In the Enter the object names to select box, enter NETWORK SERVICE.
  - NETWORK SERVICE running on a different computer from the one that contains the drop folder:
    - 1. Choose Object Types.

The Object Types dialog box appears.

- 2. Select the Computers check box, and then choose OK.
- 3. Verify that the location is correct.
- 4. In the Enter the object names to select box, enter BuildMachine\$ where BuildMachine is the name of the build machine on which the build agent is running.
- A domain account: Verify that the location is correct. In the Enter the object names to select box, enter the name of the account.
- 9. Choose OK.
- 10. Repeat the previous two steps for the build service account on the build

server that hosts your build agents.

- 11. In the Permissions for FolderName dialog box, choose the account that you just added to the Group or user names list.
- 12. Select the Change and Read check boxes, and then choose OK.

# **Drop outputs into TFS when you use the Hosted Build Controller**

When you use Visual Studio 2012 Update 1 (or a newer version) to create a build definition and the Hosted Build Controller to run it, you should drop the outputs into your Team Foundation Service server.



#### Caution

Although the system accepts a version control drop folder as the staging location when you use

<u>Team Foundation Service</u>, we do not recommend this approach.

If you do use a version control drop folder, you must designate a folder that is intended only for use as a drop folder. This folder must not contain any of your team's code or other valuable source files. Otherwise, you risk destroying your source files.

To avoid degrading performance by downloading unnecessary files, make sure

your dev machine workspace mappings and your build definition workspace mappings exclude the version control drop folder.

#### ⊠Note

If you use the version of Visual Studio 2012 released before Visual Studio 2012 Update 1 to edit the build definition, the Copy build output to the server option will be disabled.
## **Next Steps**

Once you have identified and set up the drop folder, you can specify the path to this folder when you define your build and when you queue a build.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Scale out Your Build System
Send Feedback

To use Team Foundation Build for automated building and testing of your app, you must first set up a build server, add a build controller and a few build agents, and finally designate a drop folder. If you have a small start-up team working on a new project, you can probably deploy all these build system components on a single computer in a few minutes. As your team and your code base grow, you can expand your build system incrementally, with relative ease.

### ĭTip

If your team project collection is hosted on Team Foundation Service, you might be able to skip all these steps, and use the <u>Hosted Build Controller</u> instead, as explained below.

Here are some examples that demonstrate how you can start small and simple, and then later scale out the build system as your requirements become more demanding.

- Team Foundation Service with Hosted Build Controller
- Team Foundation Service with on-premises build servers
- Build system for trial usage or a very small team
- Build system for a small team
- <u>Multiple-build-server systems</u>
- Build system to support multiple team project collections

## • <u>Next Steps</u>

## Team Foundation Service with Hosted Build Controller

If your team project collection is hosted on

<u>Team Foundation Service</u> you might be able to use the Hosted Build Controller instead of deploying your own build servers.



See Use the Hosted Build Controller in a team project collection hosted on Team Foundation Service.

## ■Team Foundation Service with onpremises build servers

If your team project collection is hosted on Team Foundation Service and your team needs larger scale or customized build agents, you can connect your onpremises build servers to Team Foundation Service.



# Build system for trial usage or a very small team

If you are using Team Foundation Server on a trial basis or work on a very small team, the following topology might work for you.



This topology might work for a team that runs builds infrequently and only during off-hours, such as a team that runs only a single nightly build. However, for many teams, it is insufficient because:

- The build agent places heavy demands on the processor, which could significantly decrease the performance of the application tier.
- The build controller can exert pressure on the system's memory, especially if the controller is managing many active build agents at the same time.
- Installing Team Foundation Build Service increases the attack surface of the computer. See Build Server: Understand security risks.

# Build system for a small team

If you work on a small team with an on-premises Team Foundation Server, consider this topology:



Because build agents perform the processor-intensive work on a separate machine, they do not affect the performance of the application tier when builds are run.

You could also run the build controller on the dedicated build server. However, the topology in the illustration has the advantage of making build system changes less disruptive, such as when you must repair or replace the build server.

# **Multiple-build-server systems**

As the size of your team and your code base increases, you can incrementally add resources to meet your requirements. For example, you could add an additional controller and build agents.



The presence of Build Controller A on the same machine as the application tier is generally not a problem from a processor standpoint. However, you might move the build controller to another server because of the memory pressure and attack surface issues mentioned previously.

By using multiple build servers, you can dedicate each server to a different purpose, as described in the following examples:

- A build server on a high-performance computer dedicated to build agents that process continuous integration or gated check-in builds. The team needs these kinds of builds-especially gated check-in builds-to run quickly so that their work is not held up waiting for a build.
- A build server dedicated to nightly scheduled BVT builds that require a lot

of time to run processes such as large test runs and code analysis.

• A build server prepared and dedicated to specialized tasks such as building and testing a Windows Store app.

### ĭTip

In scenarios such as these you can apply tags to specialized build agents and then constrain your build definitions to use only build agents with the correct set of tags. See Assign tags to represent build agent capabilities or purposes, Specify Which Build Agents Process Your Build for a simple default build process, and Run activities on the build agent for an advanced custom build process.

# Build system to support multiple team project collections

The following build system topology example could support an enterprise-level software effort.



Each team project collection must have its own build controller, as shown in above. Notice how this topology isolates the build servers. Team members who work on Team Project Collection A can use only the build agents that Build Controller A controls. This constraint could be useful in situations where you need to tightly control access to more sensitive intellectual property.

## **Next Steps**

Deploy and work with a build server

To use Team Foundation Build with an on-premises Team Foundation Server, you must deploy at least one build server. You can also connect one or more on-premises build servers to Team Foundation Service.

#### ĭTip

As you scale out your system, you can replace an existing build server when you deploy a new build server. For example, you might want to host the same configuration and set of build controllers and build agents on a new, more powerful computer. See Install Team Foundation Build Service.

Deploy and configure a build controller

Use a build controller to pool one or more build agents. You can host one build controller on a build server.

Deploy and configure build agents

Use a build agent to do the processor-intensive work of your build, includes getting files from version control, provisioning the workspace, compiling the code, and running tests.

Set up drop folders

You can prepare and then designate one or more drop folders so that your build system can deliver binaries, test results, and log files to your team.

#### Manage your build system

After you deploy your build server, you can manage it from the Team Foundation Administration Console. You can manage the build controller and build agents from either Team Foundation Administration Console or from Visual Studio. Use Team Foundation Build

With your build system in place, your team is ready to create a simple build process (for example, a continuous integration build) and benefit from automated building and testing of your app.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Manage Your Build System
Send Feedback

You've configured your build servers, including your build controllers and build agents. You've defined build processes to meet your team's needs. Your team is now running and managing your automated builds to compile and test your app.

Occasionally, you will run into situations in which you need to monitor and manage your build system, such as when you are diagnosing problems or scaling out your build system.

**Required** permissions

You must be a member of the Windows Administrators group on the build server, and a member of the Project Collection Build Administrators group on your team project collection. See <u>Team Foundation Server Permissions</u>.

What do you want to do?

- <u>Check the status, manage, and modify your controllers and agents from</u> <u>Visual Studio</u>
- <u>Check the status, manage, and modify your server, controller, and agents</u> <u>from the administration console</u>
- Start, stop, or restart a build server
- <u>Restart a build controller or build agent</u>
- Enable or disable a controller or agent
- Specify the maximum number of concurrently running builds for a controller

- <u>Register or unregister a build server</u>
- Monitor the health of your build server
- <u>Use event logs to diagnose problems</u>
- <u>Take next steps</u>

## Check the status, manage, and modify your controllers and agents from Visual Studio

Sometimes you want to check the status, and manage your build controllers and build agents, for example, when a queued build process is not starting or is failing to make progress. You can do all this directly from Visual Studio.

ame 🔺	Status	State	Date updated	Properties
efault Controller - FABRIKAM-1	Available	Enabled	7/12/2012 2:35 PM	Remove
Default Agent - FABRIKAM-1	Available	Enabled	7/12/2012 2:36 PM	
Default Agent1 - FABRIKAM-1	Available	Enabled	7/12/2012 3:36 PM	Refresh
FABRIKAM-3 - Agent1	Available	Enabled	7/12/2012 3:37 PM	
efault Controller - FABRIKAM-3	Available	Enabled	7/12/2012 9:33 AM	
Default Agent - fabrikam-8rp	Available	Enabled	7/12/2012 1:56 PM	

- 1. In Visual Studio, in Team Explorer:
  - 1. If you are not already connected to a team project in the team project collection, then connect to the team project.
  - 2. Choose 🖆 Home, and then choose 🚔 Builds.
- 2. On the Builds page, choose Actions, and then Manage Build Controllers.

#### Next Steps

Check the status of your controllers and agents

• If the State is Available and Status is Enabled, then the component is probably functioning correctly.

- If the State is Disabled, you can select the component, choose Properties, and then re-enable the component.
- If the Status is Offline, then the component is disconnected. Log on to the build server and make sure it is registered and can connect to your Team Foundation Server.

View, modify, enable, or disable a controller or agent

Select it and choose Properties, and then see

<u>Deploy and Configure a Build Controller</u> or <u>Deploy and Configure Build</u> <u>Agents</u>.

Remove a controller or agent

See Remove a build controller or Remove a build agent.

Manage or resolve problems with a build server, controller or agent

See the sections below for more information.

## Check the status, manage, and modify your server, controller, and agents from the administration console

When you want to check or manage your build system components, you sometimes need the additional capabilities that are available when you log on to the build server.



- 1. Log on to the build server that you want to configure.
- 2. From Windows Start, run Team Foundation Administration Console.

The Team Foundation Administration Console appears.

- 3. In the tree pane, expand the name of the server.
- 4. Choose the Build Configuration node.

#### ⊠Note

If the message () Configure Installed Features appears instead of a view such as the one above, see Deploy a build server.

#### Next Steps

Check the status of your controllers and agents

• If any of the build controllers or build agents are processing a build, you can see messages about this ongoing work in real time from the

#### Team Foundation Administration Console.



For more information about monitoring a running build process, see Run, Monitor, and Manage Builds and Diagnose Build Problems.

• If a controller or agent is red, then the component is disconnected. Make sure your build server can connect to your Team Foundation Server.

Work with your build server



- Choose Restart, Stop, or Start to start, stop, or restart the build server.
- Choose Properties to deploy and configure a build server.
- Choose Register or Unregister to register or unregister a build server.
- Choose Events... to <u>use event logs to diagnose problems</u>.
- Monitor the health of your build server.

Work with your build controller

Default Controller - FABRIKAM-1 - Ready Controller - Properties | Delete | Disable | Restart

- Choose Properties and see <u>Deploy and Configure a Build Controller</u>.
- Choose Restart to <u>restart the controller</u>.
- Choose Enable or Disable to <u>enable or disable the controller</u>.

Work with your build agent



- Choose Properties and see <u>Deploy and Configure Build Agents</u>.
- Choose Restart to <u>restart the agent</u>.
- Choose Enable or Disable to <u>enable or disable the agent</u>.

## **Start, stop, or restart a build server**

It is a good idea to stop the build server whenever you are performing maintenance on it. From the

<u>Team Foundation Administration Console</u>, choose Stop to take all build controllers and build agents hosted on the build server offline. Choose Start when you are ready to put the controllers and agents back in service.

If the build server fails to respond (for example, if a build process fails to start or make timely progress), choosing Restart might help. When you restart the build server, the Team Foundation Build Service process is closed and re-launched into a state that is nearly as fresh as if you rebooted the computer. If builds are currently being processed by the build controller or any build agents on this build server, those builds are cancelled.

# Restart a build controller or build agent

When restarting the build server is too disruptive or unnecessary, you can choose the Restart link of the build controller or the build agent from the

<u>Team Foundation Administration Console</u>. When you restart a build controller, it reloads assemblies, and all builds it is processing are aborted. When you restart a build agent, it discards the app domain, reloads assemblies, and the build that it is processing is aborted.

# ■Enable or disable a controller or agent

You can disable a controller or basen agent to prevent builds from being assigned to it. If you disable a controller or agent that is processing a build, the build completed before the component is disabled.

This capability can help you tune or temporarily modify the way your build system works. For example, you suspect that a build server is running too many build agents because builds are being processed too slowly. You can disable some build agents and assess how this affects performance. You can then later re-enable one or more build agents as necessary.

To enable or disable a build agent or build controller:

• From the

Visual Studio Manage Build Controllers dialog box, select the build controller or build agent that you want to disable or enable, and then choose Properties. Either the Build Controller Properties or Build Agent Properties dialog box appears. Clear or select either the Build Controller service is enabled or the Build Agent service is enabled check box.

• From the <u>Team Foundation Administration Console</u>, locate the build controller or build agent that you want to disable or enable, and then chose its Enable or Disable link.

## **Register or unregister a build server**

#### From the

Team Foundation Administration Console, choose Unregister if you want to take the build server out of your build system. When you choose Unregister, you are asked whether you want to delete any build controllers and build agents that are running on the build server:

- Choose No if you want to temporarily decommission the build server. The controllers and agents on this build server are not deleted from your team project collection and appear with a Status of Offline in the <u>Visual Studio</u> <u>Manage Build Controllers dialog box</u>. You can later choose Register and restore the build controller and build agents.
- Choose Yes if you want to delete all configuration settings for this build from the team project collection. All information about any build controller and any build agents is deleted.

If the message Build Service is not registered appears, you can choose Register to re-commission the build server and enable it to host build controllers and build agents.

#### ĭTip

You can also replace an existing build server when you deploy a new build server. For example, you might want to host the same configuration and set of build controllers and build agents on a new, more powerful computer. See Configure Team Foundation Build Service Using the Team Foundation Server Configuration Tool.

# Monitor the health of your build server

While logged on to the build server, you can confirm Team Foundation Build Service is running, get information about the resources it is consuming, and confirm the general health of the build server.

- 1. Run Windows Task Manager (Task Manager on Windows 8).
- 2. On Windows 8, if the More details link appears, choose it.
- 3. Choose the Process tab.
- 4. On versions of Windows other than Windows 8, make sure Show processes from all users is selected.
- 5. On what version of Windows is your build server running?
  - Windows 8: Locate the Visual Studio Team Foundation Build Service Host process. It should be located in the Background processes section, or if your build server is running in interactive mode, in the Apps section. Observe the CPU, memory, disk, and network resources that the process is consuming.
  - Another Windows version: Locate the TFSBuildServiceHost.exe process. Observe the CPU and memory resources that the process is consuming.
- 6. Use the other tabs in Task Manager to confirm the general health of the build server. For example, you can choose the Performance tab to confirm the computer has sufficient processor and memory resources. You can then choose Resource Monitor (on Windows 8, Open Resource Monitor).

# **Use event logs to diagnose problems**

Use the Event Viewer to get information that can help you monitor how your build server is operating and to diagnose problems with your build server or your build process.

#### **View recent events**

Begin by checking the most recent events. From the

<u>Team Foundation Administration Console</u>, choose Events... to open Event Viewer to display the most recent informational, warning, and error messages.

### **View the operational log**

To get more detailed information over a longer period of time, in the Event Viewer, in the tree pane expand Applications and Services Logs, Microsoft, Team Foundation Server, Build-Services, and then choose Operational.

## **Wiew the analytic log**

In most cases you don't need to read the internal messages exposed by the analytic log. However, this data may be useful when you are working with customer support to resolve an issue.

## To enable the analytic log

- 1. In Event Viewer, choose View, Show Analytic and Debug Logs.
- 2. In the tree pane, expand Applications and Services Logs, Microsoft, Team Foundation Server, Build-Services, and then choose Analytic.
- 3. While Analytic is still selected, open its shortcut menu and then choose Enable.

ĭTip

By default when this log reaches a data limit, the Event Viewer stops gathering data. To modify the limit or change this behavior, select the Analytic node, open it's shortcut menu, and choose Properties.

### **Customize and view the data**

To save the data in a permanent file that is easy to scan and view, in the tree pane, select one of the nodes described above, open its shortcut menu, and then choose Save All Events.... Use the Save As dialog box to save the data in the format that meets your needs.

## **Next Steps**

Deploy and work with a build server

To use Team Foundation Build with an on-premises Team Foundation Server, you must deploy at least one build server. The server can be deployed on a physical computer or a virtual machine.

Deploy and configure a build controller

Use a build controller to perform lightweight tasks and distribute the processor-intensive work of your build process to its pool of build agents. You can host one build controller on a build server.

Deploy and configure build agents

Use a build agent to do the processor-intensive work of your build, includes getting files from version control, provisioning the workspace, compiling the code, and running tests. You host can one or more build agents on a build server.

Set up drop folders

You can prepare and then designate one or more drop folders so that your build system can deliver binaries, test results, and log files to your team.

Scale out your Team Foundation Build system

As your team and your code base grow, you can expand your build system incrementally with relative ease.

#### Send

feedback on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Backing up and Restoring Your Deployment
See Also Send Feedback

To help prevent data loss and reduce the amount of time lost to a hardware or software failure, you can back up the databases that Team Foundation Server (TFS) depends on. If you are familiar with backing up and restoring SQL Server databases, you'll find the process of backing up and restoring TFS very familiar. All of the information required for restoring a TFS deployment is stored in SQL Server databases. You do not have to worry about backing up Team Foundation client computers or application-tier servers.

#### ĭTip

<u>Cumulative Update 2</u> for TFS 2012 includes the Scheduled Backups feature, which makes backing up and restoring your deployment much simpler.

The complexity of backing up and restoring your deployment depends largely on the topology of your deployment and on whether you are restoring data to existing hardware, or if you are restoring a deployment after hardware failure. It also depends on whether you are backing up and restoring part of your deployment instead of the entire deployment. For example, if you want to back up an individual team project collection and then restore it to a different deployment of TFS, you must first remember to detach the team project collection before backing up the database. Otherwise, you will not be able to attach the team project collection to the new deployment. Backing up and restoring a deployment is also made more complex if you have extended your deployment with SQL Server Reporting Services or SharePoint Products. In addition to the TFS databases, you must also back up and restore the databases in those programs that are associated with your deployment. If one of the servers in your deployment experiences a hardware failure or otherwise becomes unavailable, you can use the procedures in this section to help restore your deployment to an operational state. You will also need to reinstall software on new servers before you restore the databases. A tutorial is available specifically to assist administrators of single-server deployments to restore to new hardware, but the information in that tutorial should be useful for any administrator restoring a deployment. For more information, see <u>Restoring a Single-Server Deployment to New Hardware [Tutorial]</u>.

## **Common Tasks**

**Common Tasks** 

Supporting Content

Determine the best backup strategy for your deployment: You can Understanding review the types of Backing Up backups available for Team your data and choose a Foundation backup strategy that Server best suits your business needs.

Back up your data: You can schedule jobs to back up the databases that TFS uses. If your deployment uses reporting, you must also back up the encryption Create Back **Up Schedule** key for reporting services. If you use the and Plan Scheduled Backups features to back up your deployment, you will back up the encryption key as part of that process.

Restoring a

Restore databases from <u>Single-Server</u> backup: You can restore **Deployment** the databases that TFS to New uses to the same server Hardware [Tutorial] or servers on which they were installed, or Restore the you can restore them to **Databases** a different server or servers.

**Restore Data** to the Same Location

Recover from a hardware failure: If your application-tier server has a hardware failure, you can install software and configure a new server to act as the application-tier server. If your data-tier server has a hardware failure, you can install software and restore the databases that TFS uses to the new server. If the hardware that supports your single-server deployment fails, you can restore it to a new server.

Restore an **Application-Tier Server** 

**Restoring** a **Single-Server** Deployment to New Hardware [Tutorial]

Back up and restore Lab Management components: In this release, backing up and Restore Lab restoring Visual Studio Lab Management requires additional steps.

## See Also

#### Concepts

<u>Organizing Your Server with Team Project Collections</u> <u>Team Foundation Server Architecture</u>

#### **Other Resources**

Team Foundation Server Services and Service Accounts Managing Permissions

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Understanding Backing Up Team Foundation Server
See Also Send Feedback

You can help protect your deployment from data loss by creating a regular schedule of backups for the databases that Visual Studio Team Foundation Server (TFS) depends on. To restore your deployment of Team Foundation in its entirety, you must first back up all databases for TFS. If your deployment includes SharePoint Products or SQL Server Reporting Services, you must also back up the databases that TFS uses within those components. To prevent synchronization errors or data mismatch errors, you must synchronize all backups to the same time stamp. The easiest way to ensure successful synchronization is by using marked transactions. By routinely marking related transactions in every Team Foundation database, you establish a series of common recovery points in the databases. For step-by-step guidance for backing up a single-server deployment that uses SharePoint Foundation 2010 and also uses reporting, see <u>Create Back Up Schedule and Plan</u>.

In addition, if you intend to restore a team project collection database to a different deployment from which it originated, you must first detach the database from Team Foundation Server before you back it up. For more information, see <u>Move a Team Project Collection</u>.

### ĭTip

<u>Cumulative Update 2</u> for TFS 2012 includes the Scheduled Backups feature. The tools provided by this feature make backing up and restoring your deployment much simpler.

# **Backing Up Databases**

You can create backups of the databases that TFS uses as a key aspect of protecting your deployment of Team Foundation against loss. The following table and accompanying illustrations show which databases you must back up and provide examples of how those databases might be physically distributed in a deployment.

Database Type	Product	Optional component?
Configuration database	Team Foundation Server	No
Warehouse database	Team Foundation Server	No
Team project collection databases	Team Foundation Server	No
SharePoint Products databases	SharePoint Products	Yes
Reporting databases	SQL Server Reporting Services	Yes

Analysis databases	SQL Server Analysis Services	Yes

## **Deployment Topologies**

Based on your deployment configuration, all the databases that require backing up might be on the same physical server, as in this example topology. This example does not include SharePoint Products or Reporting Services, so you do not have to back up any databases that are associated with reporting, analysis, or SharePoint Products.



As an alternative, the databases might be distributed across many servers and server farms. In this example topology, you must back up the following databases, which are scaled across six servers or server farms:

- the configuration database
- the warehouse database
- the team project collection databases that are located on the SQL Server cluster
- the collection database that is located on the stand-alone server that is running SQL Server
- the SharePoint Products administrative databases and the site collection
databases for both SharePoint Web applications<sup>1</sup>

- the databases that are located on the server that is running Reporting Services
- the database that is located on the server that is running Analysis Services

<sup>1</sup> If your SharePoint databases are scaled across multiple servers, you cannot use the Scheduled Backups feature to back them up. You will have to manually configure backups for those databases, and ensure that those backups are synchronized with the backups for TFS databases. See <u>Manually Back Up Team</u> <u>Foundation Server</u> for more details.



In both of these examples, you do not have to back up any of the clients that connect to the server. However, you might need to manually clear the caches for Team Foundation Server on the client computers before they can reconnect to the restored deployment.

### **Databases to Back Up**

The following list provides additional detail about what you must back up, depending on your deployment resources.

### **▲**Caution

All the databases in the following list are SQL Server databases. Although you can use SQL Server Management Studio to back up individual databases at any time, you should avoid using such individual backups when possible. You might experience unexpected results if you restore from individual backups because the databases that TFS uses are all related. If you back up only one database, the data in that database will be out of synchronization with the data in the other databases.

• Databases for Team Foundation Server The logical data tier for Team Foundation Server includes several SQL Server databases, including the configuration database, the warehouse database, and a database for each team project collection in the deployment. These databases might all be on the same server, distributed across several instances in the same SQL Server deployment, or distributed across multiple servers. Regardless of their physical distribution, you must back up all of the databases to the same time stamp to help ensure against data loss. You can perform database backups manually or automatically by using maintenance plans that run at specific times or intervals.

### **A**Caution

The list of databases for Team Foundation Server is not static. A new database is created every time you create a collection. When you create a collection, make sure that you add the database for that collection to your maintenance plan.

• Databases for SharePoint Products If your deployment uses SharePoint

Products to host team project portals, you must back up several databases. These databases include the administration database for each SharePoint Web application that your deployment uses and the site collection databases that host team project portals. Ideally, your deployment has been configured to use a separate site collection for each team project collection in your deployment. Just as team project collections can be backed up and restored as a unit in Team Foundation Server, site collections can be backed up and restored in SharePoint Products. If one or more collections in your deployment are using sites or sub-sites instead of site collections as their root site, you might not be able to fully back up and restore the collections. For more information, see <u>Organizing Your Server with Team Project Collections</u>.

#### ⊠Note

You might assume that you must back up both databases and Web sites for the team project portal pages. However, SharePoint Products dynamically generates the web sites from the databases. So when you back up the databases, you also back up the sections of the team project that appear as web sites. If you have created custom site collections, site templates, or web parts in SharePoint Products but outside Team Foundation, you must back them up separately. For more information, see the following page on the Microsoft web site: <u>Backup (SharePoint Foundation 2010</u>).

- Databases for Reporting Services and Analysis Services If your deployment uses SQL Server Reporting Services or SQL Server Analysis Services to generate reports for Team Foundation Server, you must back up the reporting and analysis databases. However, you must still regenerate certain databases after restoration, such as the warehouse.
- Encryption key for the report server The report server has an encryption key that you must back up. This key safeguards sensitive information that is stored in the database for the report server. You can manually back up this key by using either the Reporting Services Configuration tool or a command-line tool.

# **Advanced Preparation for Backups**

When you deploy Team Foundation, you should keep a record of the accounts that you create and any computer names, passwords, and setup options that you specify. You should also keep a copy of all recovery materials, documents, and database and transaction log backups at a secure location. To safeguard against a disaster, such as a fire or an earthquake, you should keep duplicates of your server backups in a different location from the location of the servers. This strategy will help protect you against the loss of critical data. As a best practice, you should keep three copies of the backup media, and you should keep at least one copy offsite in a controlled environment.

#### **▲Important**

Perform a trial data restoration periodically to verify that your files are correctly backed up. A trial restoration can reveal hardware problems that do not appear with software verifications.

When you back up and restore a database, you must back up the data onto media with a network address (for example, tapes and disks that have been shared as network drives). Your backup plan should include provisions for managing media, such as the following tactics:

- A tracking and management plan for storing and recycling backup sets.
- A schedule for overwriting backup media.
- In a multi-server environment, a decision to use either centralized or distributed backups.
- A way of tracking the useful life of media.
- A procedure to minimize the effects of the loss of a backup set or backup media (for example, a tape).
- A decision to store backup sets onsite or offsite and an analysis of how this

decision might affect recovery time.

Because data for Team Foundation is stored in SQL Server databases, you do not have to back up the computers on which clients of Team Foundation are installed. If a media failure or disaster that involved those computers were to occur, you can reinstall the client software and reconnect to the server. By reinstalling the client software, your users will have a cleaner and more reliable alternative to restoring a client computer from a backup.

You can back up a server by using the Scheduled Backups features available in Update 2, or by manually creating maintenance plans in SQL Server to back up the databases that relate to your deployment of Team Foundation. The databases for Team Foundation Server work in relationship with one other, and if you create a manual plan, you should back them up and restore them at the same time. For more information about strategies for backing up databases, see the following page on the Microsoft Web site: Introduction to Backup and Restore Strategies in SQL Server.

### **Types of Backups**

Understanding the types of backups available can help you determine the best options for backing up your deployment. For example, if you are working with a large deployment and want to protect against data loss while efficiently using limited storage resources, you can configure differential backups as well as full data backups. If you are using SQL Server ALwaysOn, you can take backups of your secondary database. You can also try using backup compression or splitting backups across multiple files. Here are brief descriptions of your backup options:

Full Data Backups (Databases) A full database backup is necessary for the recoverability of your deployment. A full backup includes part of the transaction log so that you can recover the full backup. Full backups are self-contained in that they represent the entire database as it existed when you backed it up. For more information, see the following page on the Microsoft Web site: Full Database Backups.

Differential Data Backups (Databases) A differential database backup records only the data that has changed since the last full database backup, which is called the differential base. Differential database backups are smaller and faster than full database backups. This option saves backup time at the cost of increased complexity. For large databases, differential backups can occur at shorter intervals than database backups, which reduces the work-loss exposure. For more information, see the following page on the Microsoft Web site: <u>Differential Database Backups</u>.

You should also back up your transaction logs regularly. These backups are necessary for recovering data when you use the full database backup model. If you back up transaction logs, you can recover the database to the point of failure or to another specific point in time.

Transaction Log Backups The transaction log is a serial record of all modifications that have occurred in a database in addition to the transaction that performed each modification. The transaction log records the start of each transaction, the changes to the data, and, if necessary, enough information to undo the modifications made during that transaction. The log grows continuously as logged operations occur in the database.

By backing up transaction logs, you can recover the database to an earlier point in time. For example, you can restore the database to a point before unwanted data was entered or to a point of failure. Besides database backups, transaction log backups must be part of your recovery strategy. For more information, see the following page on the Microsoft Web site: <u>Working with Transaction Log Backups</u>.

Transaction log backups generally use fewer resources than full backups. Therefore, you can create transaction log backups more frequently than full backups, which reduces your risk of losing data. However, sometimes a transaction log backup is larger than a full backup. For example, a database might have a high transaction rate, which causes the transaction log to grow quickly. In this situation, you should create transaction log backups more frequently. For more information, see the following page on the Microsoft Web site: <u>Troubleshooting a Full Transaction Log</u>.

You can perform the following types of transaction log backups:

- A pure log backup contains only transaction log records for an interval, without any bulk changes.
- A bulk log backup contains log and data pages that were changed by bulk operations. Point-in-time recovery is not allowed.

• A tail-log backup is taken from a possibly damaged database to capture the log records that have not yet been backed up. A tail-log backup is taken after a failure to prevent work loss and can contain either pure log or bulk log data.

Because synchronization of data is critical for successful restoration of Team Foundation Server, you should use marked transactions as part of your backup strategy if you are configuring backups manually. For more information, see <u>Create Back Up Schedule and Plan</u> and <u>Manually Back Up Team Foundation</u> <u>Server</u> (2010).

# Backups for Services (Application Tier)

The only backup that you must perform for the logical application tier is for the encryption key for Reporting Services. If you use the Scheduled Backups feature to back up your deployment, this key will be backed up for you as part of the plan. You might assume that you must back up Web sites that are used as team project portals. If you integrated SharePoint Products as part of your deployment of Team Foundation Server, the portals will be backed up as part of backing up the databases for Team Foundation Server and SharePoint Products. However, if you specified a web site that was not created by using an integrated web application, you must back up and restore those sites manually. In addition, if you have any customizations to SharePoint Products or to services, you must also back those up or otherwise record them so that they can be reproduced on a new server.

Although you can back up an application tier more easily than a data tier, you must perform many steps when you restore an application tier. You must install another application tier for Team Foundation, redirect team project collections to use the new application tier, and redirect the portal sites for team projects.

# **Default Database Names**

If you do not customize the names of your databases, you can use the following table to identify the databases used in your deployment of Team Foundation Server. As mentioned previously, not all deployments have all these databases. For example, if you did not configure Team Foundation Server with Reporting Services, you will not have the ReportServer or ReportServerTempDB databases. Similarly, you will not have the database for System Center Virtual Machine Manager (SCVMM), which is named VirtualManagerDB, if you did not configure Team Foundation Server to support Lab Management. In addition, the databases that Team Foundation Server uses might be distributed across more than one instance of SQL Server or across more than one server.

#### ⊠Note

By default, the prefix TFS\_ is added to the names of any databases that are created automatically when you install Team Foundation Server or while it is operating.

Database	Description
TFS_Configuration	The configuration database for Team Foundation Server contains the catalog, server names, and configuration data for the deployment. The name of this database might include additional characters between TFS_ and Configuration, such as the user name of the person who installed Team Foundation Server. For example, the name of the database might be TFS_UserNameConfiguration.

TFS_Warehouse	The warehouse database contains the data for building the warehouse that Reporting Services uses. The name of this database might include additional characters between TFS_ and Warehouse, such as the user name of the person who installed Team Foundation Server. For example, the name of the database might be TFS_UserNameWarehouse.
TFS_CollectionName	The database for a team project collection contains all data for the team projects in that collection. This data includes source code, build configurations, and lab management configurations. The number of collection databases will equal the number of collections. For example, if you have three collections in your deployment, you must back up three collection databases. The name of each database might include additional characters between TFS_ and CollectionName, such as the user name of the person who created the collection. For example, the name of a collection database might be TFS_UserNameCollectionName.

T	FS_Analysis	The database for SQL Server Analysis Services contains the data sources and cubes for your deployment of Team Foundation Server. The name of this database might include additional characters between TFS_ and Analysis, such as the user name of the person who installed Analysis Services. For example, the name of the database might be TFS_UserNameAnalysis. <b>Note</b> You can back up this database, but you must rebuild the warehouse from the restored TFS_Warehouse database.
		The database for Reporting Services contains the reports and report settings for your deployment of Team Foundation Server.
		⊠Note
R	eportServer	If Reporting Services is installed on a separate server from Team Foundation Server, this database might not be present on the data-tier server for Team Foundation. In that case, you must configure, back up, and restore it separately from Team Foundation Server.

However, you should synchronize the maintenance of the databases to avoid synchronization errors.

The temporary database for Reporting Services temporarily stores information when you run specific reports.

#### ĭNote

ReportServerTempDB ReportServerTempDB If Reporting Services is installed on a separate server from Team Foundation Server, this database might not be present on the data-tier server for Team Foundation. In that case, you must configure, back up, and restore it separately from Team Foundation Server. However, you should synchronize the maintenance of the databases to avoid synchronization errors.

> The configuration database for SharePoint Products contains a list of all sites, such as content databases, site templates, custom web parts, and other settings from SharePoint Central Administration.

⊠Note

WSS_Config	You should not use marked transactions if you manually back up or restore the databases that SharePoint Products uses. However, to help avoid synchronization errors, you should attempt to synchronize your backup and restoration schedules for SharePoint Products and Team Foundation Server. For more information, see Create a backup plan for SharePoint Foundation 2010.
	The content database for SharePoint Products contains the actual content in team project portals. <b>Note</b>
WSS_Content	The name of this database will vary based on the version of SharePoint Products that is installed and whether the person who installed it customized the name. You should not use marked transactions if you manually back up or restore the databases that SharePoint Products uses. However, to help avoid synchronization errors, you should attempt to synchronize your backup and restoration schedules for SharePoint Products and Team Foundation Server. For more

information, see Create a backup plan for SharePoint Foundation 2010.

The administration database for SharePoint Products contains the security information for users, roles, and databases.

#### ⊠Note

WSS\_AdminContent
You should not use marked transactions if you manually back up or restore the databases that SharePoint Products uses. However, to help avoid synchronization errors, you should attempt to synchronize your backup and restoration schedules for SharePoint Products and Team Foundation Server. For more information, see Create a backup plan for SharePoint Foundation 2010.

The administration database for SCVMM contains the information that you view in the SCVMM Administrator Console, such as virtual machines, virtual machine hosts, virtual machine library servers, and their properties.

⊠Note

VirtualManagerDB	If SCVMM is installed on a
VirtualiviandgerDD	separate server from Team
	Foundation Server.
	this database might not be
	present on the data-tier server
	for Team Foundation. In that
	case, you must configure, back
	up, and restore it separately
	from Team Foundation Server.
	However, you should use
	marked transactions and
	synchronize the maintenance of
	the databases to avoid
	synchronization errors.

## See Also

### Concepts

Backing up and Restoring Your Deployment

#### **Other Resources**

<u>Restoring a Single-Server Deployment to New Hardware [Tutorial]</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Manually Back Up Team Foundation Server
See Also Send Feedback

You can manually back up data for Visual Studio Team Foundation Server by using the tools that SQL Server provides. As of Cumulative Update 2, TFS includes a Scheduled Backups feature to automatically configure backups. However, you might need to configure backups manually if your deployment has security restrictions that prevent use of that tool. To manually back up Team Foundation Server, you must not only back up all databases that the deployment uses, you must also synchronize the backups to the same point in time. You can manage this synchronization most effectively if you use marked transactions. If you routinely mark related transactions in every database that Team Foundation uses, you establish a series of common recovery points in those databases. If you regularly back up those databases, you reduce the risk of losing productivity or data because of equipment failure or other unexpected events.

If your deployment uses SQL Server Reporting Services, you must back up not only the databases but also the encryption key. For more information, see How to: Back Up the Reporting Services Encryption Key.

The procedures in this topic explain how to create maintenance plans that perform either a full or an incremental backup of the databases and how to create tables and stored procedures for marked transactions. For maximum data protection, you should schedule full backups to run daily or weekly and incremental backups to run hourly. You can also back up of the transaction logs. For more information, see the following page on the Microsoft website: <u>Creating Transaction Log Backups</u>.

#### ⊠Note

Many procedures in this topic specify the use of SQL Server Management

Studio. If you installed SQL Server Express Edition, you cannot use that tool unless you download SQL Server Management Studio Express. To download this tool, see the following page on the Microsoft website: <u>Microsoft SQL Server 2008 Management Studio Express</u>.

In this topic:

- Required Permissions
- <u>Identify Databases</u>
- <u>Create Tables in Databases</u>
- <u>Create a Stored Procedure for Marking Tables</u>
- Create a Stored Procedure for Marking All Tables At Once
- <u>Create a Stored Procedure to Automatically Mark Tables</u>
- <u>Create a Scheduled Job to Run the Table-Marking Procedure</u>
- <u>Create a Maintenance Plan For Full Backups</u>
- <u>Create a Maintenance Plan For Differential Backups</u>
- <u>Create a Maintenance Plan For Transaction Backups</u>
- Back Up the Reporting Services Encryption Key
- <u>Create a Back Up Plan for SharePoint Foundation 2010</u>
- Back Up Additional Lab Management Components

### **Required Permissions**

To perform this procedure, you must be a member of all the following groups:

• The Administrators security group on the server that is running the administration console for Team Foundation.

- The SQL Server System Administrator security group. Alternatively, your SQL Server Perform Back Up and Create Maintenance Plan permissions must be set to Allow on each instance of SQL Server that hosts the databases that you want to back up.
- The Farm Administrators group in SharePoint Foundation 2010, or an account with the permissions required to back up the farm.

# **Identify Databases**

Before you begin, you should take the time to identify all the databases you will need to back up if you would ever have to fully restore your deployment. This includes databases for SharePoint Foundation 2010 and SQL Server Reporting Services. These might be on the same server, or you might have databases distributed across multiple servers. For a complete table and description of TFS databases, including the default names for the databases, see

**Understanding Backing Up Team Foundation Server.** 

### To identify databases

- 1. Open SQL Server Management Studio, and connect to the database engine.
- 2. In SQL Server Management Studio, in Object Explorer, expand the name of the server and then expand Databases.
- 3. Review the list of databases and identify those used by your deployment.

For example, Fabrikam, Inc.'s TFS deployment is a single-server configuration, and it uses the following databases:

- the configuration database (Tfs\_Configuration)
- the collection database (Tfs\_DefaultCollection)
- the database for the data warehouse (Tfs\_Warehouse)
- the reporting databases (ReportServer and ReportServerTempDB)
- the databases used by SharePoint Foundation 2010 (WSS\_AdminContent, WSS\_Config, WSS\_Content, and WSS\_Logging)

### **▲Important**

Unlike the other databases in the deployment, the databases used by SharePoint Foundation 2010 should not be backed up using the tools in SQL Server. Follow the separate procedure "<u>Create a Back Up Plan</u> for SharePoint Foundation 2010" later in this topic for backing up these databases.

## **Create tables in databases**

To make sure that all databases are restored to the same point, you can create a table in each database to mark transactions. You can use the Query function in SQL Server Management Studio to create an appropriate table in each database.

#### **▲Important**

Do not create tables in any databases that SharePoint Products uses.

### To create tables to mark related transactions in databases that Team Foundation uses

- 1. Open SQL Server Management Studio, and connect to the database engine.
- 2. In SQL Server Management Studio, highlight the name of the server, open the submenu, and then choose New Query.

The Database Engine Query Editor window opens.

3. On the Query menu, choose SQLCMD Mode.

The Query Editor executes sqlcmd statements in the context of the Query Editor. If the Query menu does not appear, select anywhere in the new query in the Database Engine Query Editor window.

4. On the SQL Editor toolbar, open the Available Databases list, and then choose TFS\_Configuration.

#### ⊠Note

TFS\_Configuration is the default name of the configuration database. This name is customizable and might vary.

5. In the query window, enter the following script to create a table in the

configuration database:

### Copy Code

Use Tfs\_Configuration Create Table Tbl\_TransactionLogMark ( logmark int ) GO Insert into Tbl\_TransactionLogMark (logmark) Values (1) GO

6. Choose the F5 key to run the script.

If the script is well-formed, the message "(1 row(s) affected.)" appears in the Query Editor.

- 7. (Optional) Save the script.
- 8. Repeat steps 4–7 for every database in your deployment of TFS, except for those used by SharePoint Products. In the fictitious Fabrikam, Inc. deployment, you would repeat this process for all of the following databases:
  - Tfs\_Warehouse
  - Tfs\_DefaultCollection
  - ReportServer
  - ReportServerTempDB

# ■Create a stored procedure for marking tables

After the tables have been created in each database that you want to back up, you must create a procedure for marking the tables.

# To create a stored procedure to mark transactions in each database that Team Foundation Server uses

- 1. In SQL Server Management Studio, open a query window, and make sure that SQLCMD Mode is turned on.
- 2. On the SQL Editor toolbar, open the Available Databases list, and then choose TFS\_Configuration.
- 3. In the query window, enter the following script to create a stored procedure to mark transactions in the configuration database:

### Copy Code

```
Create PROCEDURE sp_SetTransactionLogMark
@name nvarchar (128)
AS
BEGIN TRANSACTION @name WITH MARK
UPDATE Tfs_Configuration.dbo.Tbl_TransactionLogMark SET logmark
COMMIT TRANSACTION
GO
```

4. Choose the F5 key to run the procedure.

If the procedure is well-formed, the message "Command(s) completed successfully." appears in the Query Editor.

- 5. (Optional) Save the procedure.
- 6. Repeat steps 2–5 for every database in your deployment of TFS. In the

Fabrikam, Inc. deployment, the administrator, Jill, repeats this process for all of the following databases:

- Tfs\_Warehouse
- Tfs\_DefaultCollection
- ReportServer
- ReportServerTempDB

### ĭTip

Make sure that you select the name of the database you want to create the stored procedure for from the Available Database list in Object Explorer before you create the procedure. Otherwise when you run the script the command will display an error that the stored procedure was already exists.

# ■Create a stored procedure for marking all tables at once

To make sure that all databases are marked, you can create a procedure that will run all the procedures that you just created for marking the tables. Unlike the previous procedures, this procedure runs only in the configuration database.

# To create a stored procedure that will run all stored procedures for marking tables

- 1. In SQL Server Management Studio, open a query window, and make sure that SQLCMD Mode is turned on.
- 2. On the SQL Editor toolbar, open the Available Databases list, and then choose TFS\_Configuration.
- 3. In the query window, create a stored procedure that executes the stored procedures that you created in each database that TFS uses. Replace ServerName with the name of the server that is running SQL Server, and replace Tfs\_CollectionName with the name of the database for each team project collection.

In the example deployment, the name of the server is FABRIKAMPRIME, and there is only one team project collection in the deployment, the default one created when she installed Team Foundation Server (DefaultCollection). With that in mind, Jill creates the following script:

### Copy Code

CREATE PROCEDURE sp\_SetTransactionLogMarkAll @name nvarchar (128) AS BEGIN TRANSACTION EXEC [FABRIKAMPRIME].Tfs\_Configuration.dbo.sp\_SetTransactionLogI EXEC [FABRIKAMPRIME].ReportServer.dbo.sp\_SetTransactionLogMark ( EXEC [FABRIKAMPRIME].ReportServerTempDB.dbo.sp\_SetTransactionLog EXEC [FABRIKAMPRIME].Tfs\_DefaultCollection.dbo.sp\_SetTransactionLog

```
EXEC [FABRIKAMPRIME].Tfs_Warehouse.dbo.sp_SetTransactionLogMark
COMMIT TRANSACTION
GO
```

4. Choose the F5 key to run the procedure.

#### ⊠Note

If you have not restarted SQL Server Management Studio since you created the stored procedures for marking transactions, one or more red wavy lines might underscore the name of the server and the names of the databases. However, the procedure should still run.

If the procedure is well-formed, the message "Command(s) completed successfully." appears in the Query Editor.

5. (Optional) Save the procedure.

# ■Create a stored procedure to automatically mark tables

When you have a procedure that will run all stored procedures for table marking, you must create a procedure that will mark all tables with the same transaction marker. You will use this marker to restore all databases to the same point.

# To create a stored procedure to mark the tables in each database that Team Foundation Server uses

- 1. In SQL Server Management Studio, open a query window, and make sure that SQLCMD Mode is turned on.
- 2. On the SQL Editor toolbar, open the Available Databases list, and then choose TFS\_Configuration.
- 3. In the query window, enter the following script to mark the tables with 'TFSMark':

### Copy Code

```
EXEC sp_SetTransactionLogMarkAll 'TFSMark'
G0
```

#### ⊠Note

TFSMark is an example of a mark. You can use any sequence of supported letters and numbers in your mark. If you have more than one marked table in the databases, record which mark you will use to restore the databases. For more information, see the following page on the Microsoft website:

Using Marked Transactions.

• Choose the F5 key to run the procedure.

If the procedure is well-formed, the message "(1 row(s) affected)" appears in the Query Editor. The WITH MARK option applies only to the first "BEGIN TRAN WITH MARK" statement for each table that has been marked.

• Save the procedure.

# ■Create a scheduled job to run the table-marking procedure

Now that you have created and stored all the procedures that you need, you must schedule the table-marking procedure to run just before the scheduled backups of the databases. You should schedule this job to run approximately one minute before the maintenance plan for the databases runs.

### To create a scheduled job for table marking in SQL Server Management Studio

1. In Object Explorer, expand SQL Server Agent, open the Jobs menu, and then choose New Job.

The New Job window opens.

- 2. In Name, specify a name for the job. For example, Jill types the name "MarkTableJob" for her job name.
- 3. (Optional) In Description, specify a description of the job.
- 4. In Select a page, choose Steps and then choose New.
- 5. The New Job Step window opens.
- 6. In Step Name, specify a name for the step.
- 7. In Database, choose the name of the configuration database. For example, Jill's deployment uses the default name for that database, TFS\_Configuration, so she chooses that database from the drop-down list.
- 8. Choose Open, browse to the procedure that you created for marking the tables, choose Open two times, and then choose OK.

⊠Note

The procedure that you created for marking the tables runs the following step:

### Copy Code

EXEC sp\_SetTransactionLogMarkAll 'TFSMark'

9. In Select a page, choose Schedules, and then choose New.

The New Job Schedule window opens.

- 10. In Name, specify a name for the schedule.
- 11. In Frequency, change the frequency to match the plan that you will create for backing up the databases. In the example deployment, Jill wants to run incremental backups daily at 2 A.M., and full backups on Sunday at 4 A.M. For marking the databases for the incremental backups, she changes the value of Occurs to Daily. When she creates another job to mark the databases for the weekly full backup, she keeps the value of Occurs at Daily, and selects the Sunday check box.
- 12. In Daily Frequency, change the occurrence so that the job is scheduled to run one minute before the backup for the databases, and then choose OK. In the example deployment, in the job for the incremental backups, Jill specifies 1:59 A.M.. In the job for the full backup, Jill specifies 3:59 A.M..
- 13. In New Job, choose OK to finish creating the scheduled job.

# ■Create a maintenance plan for full backups

After you create a scheduled job for marking the databases, you can use the Maintenance Plan Wizard to schedule full backups of all of the databases that your deployment of TFS uses.

### **▲Important**

If your deployment is using the Enterprise or Datacenter editions of SQL Server, but you think you might want to restore databases to a server running Standard edition, you must use a backup set that was made with SQL Server compression disabled. Unless you disable data compression, you will not be able to successfully restore Enterprise or Datacenter edition databases to a server running Standard edition. You should turn off compression before creating your maintenance plans. To turn off compression, follow the steps in the

### Microsoft Knowledge Base article.

### To create a maintenance plan for full backups

- 1. In SQL Server Management Studio, expand the Management node, open the Maintenance Plans sub-menu, and then choose Maintenance Plan Wizard.
- 2. On the welcome page for the SQL Server Maintenance Plan Wizard, choose Next.

The Select Plan Properties page appears.

3. In the Name box, specify a name for the maintenance plan.

For example, Jill decides to create a plan for full backups named TfsFullDataBackup.

- 4. Choose Single schedule for the entire plan or no schedule, and then choose Change.
- 5. Under Frequency and Daily Frequency, specify options for your plan. For example, Jill specifies a weekly backup to occur on Sunday in Frequency, and specifies 4 A.M. in Daily Frequency.

Under Duration, leave the default value, No end date. Choose OK, and then choose Next.

- 6. On the Select Maintenance Tasks page, select the Backup Database (Full), Execute SQL Server Agent Job, and Back up Database (Transaction Log) check boxes, and then choose Next.
- 7. On the Select Maintenance Task Order page, change the order so that the full backup runs first, then the Agent job, and then the transaction log backup, and then choose Next.

For more information about this dialog box, choose the F1 key. Also, search for Maintenance Plan Wizard on the following page of the Microsoft website: <u>SQL Server Books Online</u>.

- 8. On the Define Back Up Database (Full) Task page, choose the down arrow, choose All Databases, and then choose OK.
- 9. Specify the backup options for saving the files to disk or tape, as appropriate for your deployment and resources, and then choose Next.
- 10. On the Define Execute SQL Server Agent Job Task page, select the check box for the scheduled job that you created for table marking, and then choose Next.
- 11. On the Define Back Up Database (Transaction Log) Task page, choose the down arrow, choose All Databases, and then choose OK.
- 12. Specify the backup options for saving the files to disk or tape as appropriate for your deployment and resources, and then choose Next.
- 13. On the Select Report Options page, specify report distribution options, and then choose Next two times.

14. On the Complete the Wizard page, choose Finish.

SQL Server creates the maintenance plan and backs up the databases that you specified based on the frequency that you specified.

# ■Create a maintenance plan for differential backups

You can use the Maintenance Plan Wizard to schedule differential backups for all databases that your deployment of TFS uses.

### **▲Important**

SQL Server Express does not include the Maintenance Plan Wizard. You must manually script the schedule for your differential backups. For more information, see the following topic on the Microsoft website:

How to: Create a Differential Database Backup (Transact-SQL).

### To create a maintenance plan for differential backups

- 1. Log on to the server that is running the instance of SQL Server that contains the databases that you want to back up.
- 2. Choose Start, choose All Programs, choose Microsoft SQL Server 2008, and then choose SQL Server Management Studio.
  - 1. In the Server type list, choose Database Engine.
  - 2. In the Server name and Authentication lists, choose the appropriate server and authentication scheme.
  - 3. If your instance of SQL Server requires it, in User name and Password, specify the credentials of an appropriate account.
  - 4. Choose Connect.
- 3. In SQL Server Management Studio, expand the Management node, open the sub-menu, choose Maintenance Plans, and then choose Maintenance Plan Wizard.

- 4. On the welcome page for the SQL Server Maintenance Plan Wizard, choose Next.
- 5. On the Select Plan Properties page, in the Name box, specify a name for the maintenance plan.

For example, you could name a plan for differential backups TfsDifferentialBackup.

- 6. Choose Single schedule for the entire plan or no schedule, and then choose Change.
- 7. Under Frequency and Daily Frequency, specify options for your backup plan.

Under Duration, leave the default value, No end date. Choose OK, and then choose Next.

- 8. On the Select Maintenance Tasks page, select the Back up Database (Differential) check box, and then choose Next.
- 9. On the Define Back Up Database (Differential) Task page, choose the down arrow, choose All Databases, and then choose OK.
- 10. Specify the backup options for saving the files to disk or tape as appropriate for your deployment and resources, and then choose Next.
- 11. On the Select Report Options page, specify report distribution options, and then choose Next two times.
- 12. On the Complete the Wizard page, choose Finish.

SQL Server creates the maintenance plan and backs up the databases that you specified based on the frequency that you specified.
### ■Create a maintenance plan for transaction logs

You can use the Maintenance Plan Wizard to schedule transaction log backups for all databases that your deployment of TFS uses.

### **▲Important**

SQL Server Express does not include the Maintenance Plan Wizard. You must manually script the schedule for transaction-log backups. For more information, see the following topic on the Microsoft website:

How to: Create a Transaction Log Backup (Transact-SQL).

### To create a maintenance plan for transaction log backups

- 1. Log on to the server that is running the instance of SQL Server that contains the databases that you want to back up.
- 2. Choose Start, choose All Programs, choose Microsoft SQL Server 2008, and then choose SQL Server Management Studio.
- 3. In the Server type list, choose Database Engine.
  - 1. In the Server name and Authentication lists, choose the appropriate server and authentication scheme.
  - 2. If your instance of SQL Server requires it, in User name and Password, specify the credentials of an appropriate account.
  - 3. Choose Connect.
- 4. In SQL Server Management Studio, expand the Management node, open the submenu, choose Maintenance Plans, and then choose Maintenance Plan Wizard.

5. On the welcome page for the SQL Server Maintenance Plan Wizard, choose Next.

The Select Plan Properties page appears.

6. In the Name box, specify a name for the maintenance plan.

For example, you could name a plan to back up transaction logs TfsTransactionLogBackup.

- 7. Choose Single schedule for the entire plan or no schedule, and then choose Change.
- 8. Under Frequency and Daily Frequency, specify options for your plan.

Under Duration, leave the default value, No end date.

- 9. Choose OK, and then choose Next.
- 10. On the Select Maintenance Tasks page, select the Execute SQL Server Agent Job and Back up Database (Transaction Log) check boxes, and then choose Next.
- 11. On the Select Maintenance Task Order page, change the order so that the Agent job runs before the transaction-log backup, and then choose Next.

For more information about this dialog box, choose the F1 key. Also, search for Maintenance Plan Wizard on the following page of the Microsoft website: <u>SQL Server Books Online</u>.

- 12. On the Define Execute SQL Server Agent Job Task page, select the check box for the scheduled job that you created for table marking, and then choose Next.
- 13. On the Define Back Up Database (Transaction Log) Task page, choose the down arrow, choose All Databases, and then choose OK.
- 14. Specify the backup options for saving the files to disk or tape as appropriate for your deployment and resources, and then choose Next.

- 15. On the Select Report Options page, specify report distribution options, and then choose Next two times.
- 16. On the Complete the Wizard page, choose Finish.

SQL Server creates the maintenance plan and backs up the transaction logs for the databases that you specified based on the frequency that you specified.

# ■Back up the encryption key for reporting services

You must back up the encryption key for Reporting Services as part of backing up your system. Without this encryption key, you will not be able to restore the reporting data. For a single-server deployment of TFS, you can back up the encryption key for SQL Server Reporting Services by using the Reporting Services Configuration tool. You could also choose to use the **RSKEYMGMT** command-line tool, but the configuration tool is simpler. For more information about **RSKEYMGMT**, see the following page on the Microsoft website: <u>RSKEYMGMT Utility</u>.

### To back up the encryption key by using the Reporting Services Configuration tool

1. On the server that is running Reporting Services, choose Start, point to All Programs, point to Microsoft SQL Server, point to Configuration Tools, and then choose Reporting Services Configuration Manager.

The Report Server Installation Instance Selection dialog box opens.

- 2. Specify the name of the data-tier server and the database instance, and then choose Connect.
- 3. In the navigation bar on the left side, choose Encryption Keys, and then choose Backup.

The Encryption Key Information dialog box opens.

4. In File Location, specify the location where you want to store a copy of this key.

You should consider storing this key on a separate computer from the one that is running Reporting Services.

5. In Password, specify a password for the file.

6. In Confirm Password, specify the password for the file again, and then choose OK.

### Create a backup plan for SharePoint Foundation 2010

Unlike Team Foundation Server, which uses the scheduling tools in SQL Server Management Studio, there is no built-in scheduling system for backups in SharePoint Foundation 2010, and SharePoint specifically recommends against any scripting that marks or alters its databases. To schedule backups so that they occur at the same time as the backups for TFS, SharePoint Foundation 2010 guidance recommends that you create a backup script by using Windows PowerShell, and then use Windows Task Scheduler to run the backup script at the same time as your scheduled backups of TFS databases. This will help you keep your database backups in sync.

### **▲Important**

Before proceeding with the procedures below, you should review the latest guidance for SharePoint Foundation 2010. The procedures below are based on that guidance, but might have become out of date. Always follow the latest recommendations and guidance for the version of SharePoint Products you use when managing that aspect of your deployment. For more information, see the links included with each of the procedures in this section.

### To create scripts to perform full and differential backups of the farm in SharePoint Foundation 2010

- 1. Open a text editor, such as Notepad.
- 2. In the text editor, type the following, where BackupFolder is the UNC path to a network share where you will back up your data:

### Copy Code

Backup-SPFarm -Directory BackupFolder -BackupMethod Full

### ĭTip

There are a number of other parameters you could use when backing up the farm. For more information, see

### Back up a farm and Backup-SPFarm.

• Save the script as a .PS1 file. Consider giving the file an obvious name, such as "SharePointFarmFullBackupScript.PS1" or some meaningful equivalent.

• Open a new file, and create a second backup file, only this time specifying a differential backup:

### Copy Code

Backup-SPFarm -Directory BackupFolder -BackupMethod Differential

• Save the script as a .PS1 file. Consider giving the file an obvious name, such as "SharePointFarmDiffBackupScript.PS1".

#### **▲Important**

By default, PowerShell scripts will not execute on your system unless you have changed PowerShell's execution policy to allow scripts to run. For more information, see <u>Running Windows PowerShell Scripts</u>.

After you have created your scripts, you must schedule them to execute following the same schedule and frequency as the schedule you created for backing up Team Foundation Server databases. For example, if you scheduled differential backups to execute daily at 2 A.M., and full backups to occur on Sundays at 4 A.M., you will want to follow the exact same schedule for your farm backups.

To schedule your backups, you must use Windows Task Scheduler. In addition, you must configure the tasks to run using an account with sufficient permissions to read and write to the backup location, as well as permissions to execute backups in SharePoint Foundation 2010. Generally speaking, the simplest way to

do this is to use a farm administrator account, but you can use any account as long as all of the following criteria are met:

- The account specified in Windows Task Scheduler is an administrative account.
- The account specified for the Central Administration application pool and the account you specify for running the task have read/write access to the backup location.
- The backup location is accessible from the server running SharePoint Foundation 2010, SQL Server, and Team Foundation Server.

### To schedule backups for the farm

- 1. Choose Start, choose Administrative Tools, and then choose Task Scheduler.
- 2. In the Actions pane, choose Create Task.
- 3. On the General tab, in Name, specify a name for this task, such as "Full Farm Backup." In Security options, specify the user account under which to run the task if it is not the account you are using. Then choose Run whether user is logged on or not, and select the Run with highest privileges check box.
- 4. On the Actions tab, choose New.

In the New Action window, in Action, choose Start a program. In Program/script, specify the full path and file name of the full farm backup .PS1 script you created, and then choose OK.

5. On the Triggers tab, choose New.

In the New Trigger window, in Settings, specify the schedule for performing the full backup of the farm. Make sure that this schedule exactly matches the schedule for full backups of the Team Foundation Server databases, including the recurrence schedule, and then choose OK.

6. Review all the information in the tabs, and then choose OK to create the

task for the full backup for the farm.

- 7. In the Actions pane, choose Create Task.
- 8. On the General tab, in Name, specify a name for this task, such as "Differential Farm Backup." In Security options, specify the user account under which to run the task if it is not the account you are using, choose Run whether user is logged on or not, and select the Run with highest privileges check box.
- 9. On the Actions tab, choose New.

In the New Action window, in Action, choose Start a program. In Program/script, specify the full path and file name of the differential farm backup .PS1 script you created, and then choose OK.

10. On the Triggers tab, choose New.

In the New Trigger window, in Settings, specify the schedule for performing the full backup of the farm. Make sure that this schedule exactly matches the schedule for full backups of the Team Foundation Server databases, including the recurrence schedule, and then choose OK.

- 11. Review all the information in the tabs, and then choose OK to create the task for the differential backup for the farm.
- 12. In Active Tasks, refresh the list and make sure that your new tasks are scheduled appropriately, and then close Task Scheduler. For more information about creating and scheduling tasks in Task Scheduler, see <u>Task</u> <u>Scheduler How To</u>.

Home | Prepare for Installation | Install Prerequisites and Team Foundation Server | Configure Team Foundation Server to Support Your Development Teams | Create Back Up Schedule and Plan

### ■Back Up Additional Lab Management Components

If you use Visual Studio Lab Management in your deployment of Team Foundation Server, you must also back up each machine and component that Lab Management uses. The hosts for the virtual machines and the SCVMM library servers are separate physical computers that are not backed up by default. You must specifically include them when you plan your backup and restoration strategies. The following table summarizes what you should back up whenever you back up Team Foundation Server.

Machine	Component
Server that is running System Center Virtual Machine Manager 2008 (SCVMM) R2	<ul> <li>SQL Server database (user accounts, configuration data)</li> </ul>
	<ul> <li>Virtual machines (VMs)</li> </ul>
Physical host for the virtual machines	• Templates
	<ul> <li>Host configuration data (virtual networks)</li> </ul>
	• Virtual

SCVMM library server	machines	
	• Templates	
	<ul> <li>Virtual hard disks (VHDs)</li> </ul>	
	• ISO images	

The following table contains tasks and links to procedural or conceptual information about how to back up the additional machines for an installation of Lab Management. You must perform the tasks in the order shown, without skipping any tasks.

To back up the machines that are running any SCVMM components, you must be a member of the Backup Operators group on each machine.

	Common Tasks	Detailed instructions
1.	Back up the server that is running System Center Virtual Machine Manager 2008 R2.	
2.	Back up the library servers for SCVMM.	•
3.	Back up each physical host for the virtual machines.	
Back	king Up and Restoring	the SCVMM Database

### **See Also**

Tasks

Restore Data to the Same Location

### **Other Resources**

Managing Data Managing Team Foundation Server Data-Tier Servers Managing Team Foundation Server How to: Back Up the Reporting Services Encryption Key Restore Data to a Different Server or Instance Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Restore an Application-Tier Server
See Also Send Feedback

The databases for Team Foundation store all data for your deployment of Team Foundation Server. Even if you back up the application-tier server, you will not back up any data for Team Foundation Server. However, if the hardware of an application-tier server fails, you can install another application-tier server and configure it to use the databases for your deployment. That server will then replace the offline server as the application-tier server for the deployment. If your application-tier server hosted SharePoint Products, you must also restore that software on the new hardware. For more information, see <u>Backup</u> (SharePoint Foundation 2010), Backup and Recovery (SharePoint Server 2010), or Protecting and restoring a farm (Office SharePoint Server 2007).

#### ⊠Note

After you restore an application tier to new hardware, make sure that all users, groups, and service accounts for your deployment are configured with the permissions that they require to perform necessary tasks. For example, administrators for Team Foundation must be members of the local Administrators group on the application-tier server so that they can open the administration console. For more information, see <u>Add Users to Team Projects</u>, <u>Set Administrator Permissions for Team Project Collections</u>, <u>Set Administrator Permissions for Team Foundation Server</u>, and <u>Service Accounts and Dependencies in Team Foundation Server</u>.

You can also add more than one application-tier server to a deployment of Team Foundation Server, but you must configure clients to connect to that server as a separate application tier. You cannot configure automatic load balancing between application-tier servers. For actual load balancing and transparency to clients, you must first install and configure a hardware or software device for network load balancing (NLB).

### To install and configure a server as the application-tier server

1. Stop the application pools and services that Team Foundation Server uses.

For more information, see <u>TFSServiceControl Command</u>.

2. If you are using Network Service as the service account for Team Foundation (TFSService), on the application-tier server, open a Command Prompt window, and change directories to Drive:%Program Files%\ Microsoft Team Foundation Server 11.0\Tools. At the command prompt, enter the following command:

## TfsConfig Accounts /add /account: "NT Authority\Network Service" /accountType:ApplicationTier /SQLInstance:ServerName /DatabaseName:DatabaseName

### ⊠Note

For more information, see <u>Accounts Command</u>.

- 3. Install Team Foundation Server on the new server, and start the Application-Tier Only wizard.
- 4. If you are using Visual Studio Lab Management, install the System Center Virtual Machine Manager (SCVMM) Administrator Console on the application tier, and configure it to connect to the server that is running SCVMM.

For more information, see <u>Configuring Lab Management for SCVMM</u> <u>Environments</u>.

- 5. If the computer name has changed, open the administration console for Team Foundation.
- 6. In the navigation bar, choose Application Tier, and then choose Change URLs.

The Change URLs window opens.

7. In Notification URL, specify the URL for the new application-tier server, and then choose OK.

### ⊠Note

The name of the old application-tier server will still appear in the list of application-tier servers in the administration console for Team Foundation. If you select the Filter out machines that have not connected in more than 3 days check box, the old server will disappear from the list within three days.

### **See Also**

### Tasks

**Open the Team Foundation Administration Console** 

### Concepts

Restore Lab Management Components Team Foundation Server Architecture

### **Other Resources**

Restoring a Single-Server Deployment to New Hardware [Tutorial]

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Restore Data to the Same Location
See Also Send Feedback

You can restore data from a backup to the same server and instance of SQL Server for Team Foundation from which that data was backed up. For example, you might want to restore a corrupted set of databases to the last known good state.

### ĭTip

These procedures depend upon your having installed <u>Cumulative Update 2</u> for TFS 2012. In addition, if your original deployment used the Enterprise or Datacenter editions of SQL Server, and you want to restore databases to a server running Standard edition, you must use a backup set that was made with SQL Server compression disabled. Unless you disable data compression, you will not be able to successfully restore Enterprise or Datacenter edition databases to a server running Standard edition. To turn off compression, follow the steps in the <u>Microsoft Knowledge Base article</u>.

If you want to restore data to another server or another instance of SQL Server, see <u>Restoring a Single-Server Deployment to New Hardware [Tutorial]</u>.

#### ⊠Note

If you use SharePoint Products in your deployment, when you restore data, you do not have to restore the websites that are automatically generated based on the data for each team project. The data for the team project portals is contained in the databases that you restore.

The steps to restore data to the same server or servers vary based on how Team

Foundation Server is installed and configured. For simplicity, the procedures in this topic are structured for a moderately complex deployment of Team Foundation Server, as the following illustration shows:



If your topology does not exactly match this example, you might have to adjust the steps in this procedure to follow it successfully. For example, if you have a deployment where all components are installed on a single physical server, you would perform all procedures on that server. If databases for team project collections are deployed on more than one server, you must perform the steps to restore each collection database on the appropriate server. For more information about which components might be deployed on each server, see the following topics:

- <u>Understanding Backing Up Team Foundation Server</u>
- Team Foundation Server Architecture
- Examples of Simple Topology
- Examples of Moderate Topology

• Examples of Complex Topology

In this topic

- 1. <u>Required Permissions</u>
- 2. <u>Rename Databases You Want To Restore</u>
- 3. <u>Restore Team Foundation Databases</u>
- 4. <u>Update All Service Accounts</u>
- 5. <u>Restore the Warehouse</u>
- 6. <u>Clear the Data Cache On Servers</u>
- 7. Restart Services that Team Foundation Server Uses
- 8. <u>Refresh the Data Cache on Client Computers</u>

#### **Required Permissions**

To perform this procedure, you must be a member of the following groups or have the following permissions:

- A member of the Administrators security group on the server or servers that are running the administration console for Team Foundation.
- Either a member of the SQL Server System Administrator security group or your SQL Server Perform Back Up and Create Maintenance Plan permission must be set to Allow on the instance of SQL Server that hosts the databases.
- A member of the **sysadmin** security group for the database instance for Team Foundation and for the Analysis Services instance of the warehouse database.
- An authorized user of the TFS\_Warehouse database.
- A member of the TFSEXECROLE database role.

• If the deployment uses SharePoint Products, a member of the Farm Administrators group for the farm to which the SharePoint Products databases are being restored.

In addition to these permissions, you might have to address the following requirements on a computer that is running Windows Server 2008:

- To follow a command-line procedure, you might have to open an elevated Command Prompt by clicking Start, right-clicking Command Prompt, and then clicking Run as Administrator.
- To follow a procedure that requires Internet Explorer, you might have to start it as an administrator by clicking Start, clicking All Programs, right-clicking Internet Explorer, and then clicking Run as administrator.
- To access Report Manager, reports, or websites for SQL Server Reporting Services, you might have to add these sites to the list of trusted sites in Internet Explorer or start Internet Explorer as an administrator.

For more information, see the following page on the Microsoft website: <u>User</u> <u>Account Control</u>.

### Rename Databases that You Want to Restore

Before you can use the Restore wizard to restore a database that Team Foundation Server, you must first stop it, and then rename it.

### To stop databases that Team Foundation Server uses

1. Choose Start, All Programs, Microsoft SQL Server, and then choose SQL Server Management Studio.

### ⊠Note

For more information about how to restore databases, see the following page on the Microsoft website:

Implementing Restore Scenarios for SQL Server Databases.

The Connect to Server dialog box opens.

• In Server type, choose Database Engine.

• In Server name, choose or type the name of the data-tier server and database instance, and then choose Connect.

#### ⊠Note

If SQL Server is installed on a cluster, the server name is the name of the cluster and not the computer name.

SQL Server Management Studio opens.

• Expand the Databases node to show the list of databases that make up the data tier for Team Foundation.

• Open the context menu for each database you want to restore, and take it offline.

• After the database is offline, rename the database <u>following the guidance for</u> <u>your version of SQL Server</u>. Give the database a name that indicates it is the old version of the database you will replace with the restored version. For example, you might rename TFS\_DefaultCollection to TFS\_DefaultCollection\_Old.

### **Restore Team Foundation Databases**

You can restore data for Team Foundation Server by using the Restore wizard in the administration console in TFS. The Restore wizard also restores the encryption key used for reporting.

#### ⊠Note

The Restore wizard is part of the Scheduled Backups set of features available in

<u>Cumulative Update 2</u> for TFS 2012. If you do not see that node in the administration console, make sure that you have installed that update before you proceed.

### To restore databases

1. Open the administration console for TFS and navigate to Scheduled Backups to start the Restore wizard.



2. Specify the path to the backup set and choose the set you want to use for restoration.

Team Foundation Server Restore Databases Wizard					
🌆 Restore D	atabases Wiza	ard			
Backup Sets Databases Review Readiness Checks	Select Backup Sets Point to the backup path and select the point in time you want to restore your T Server to. Note: The most recent backup has been selected by default.				
Complete	Network Backup Path:	\\fabrikambackup\TFS\	Browse		
	Date 3/14/2013 5:19:55 PM 3/14/2013 4:49:54 PM 3/14/2013 4:19:54 PM 3/14/2013 3:49:54 PM 3/14/2013 3:24:08 PM 3/14/2013 3:19:57 PM 3/14/2013 2:49:53 PM 3/14/2013 2:19:53 PM 3/14/2013 1:49:54 PM 3/14/2013 1:29:28 PM 3/14/2013 1:20:52 PM	Size (MB) 192 192 191 191 190 100 100 100 100 180 179 179 38 178			

3. Complete the wizard and restore the databases.

Team Foundation Server Restore Databases Wizard					
Restore Databases Wizard					
Backup Sets Databases Review Readiness Checks Restore Databases	Configuration Progress				
Complete	<ul> <li>Granting Required Permissions for Restore Operation</li> <li>Restoring Databases</li> <li>Saving Settings</li> <li>Success</li> <li>Configuration completed successfully. Click Next</li> </ul>	Complete Complete Complete			

### **Update All Service Accounts**

You must update the service account for Team Foundation Server (TFSService) and the data sources account (TFSReports). Even if these accounts have not changed, you must update the information to ensure that the identity and the format of the accounts are appropriate.

### To update service accounts

- 1. On the server that is running SQL Server Reporting Services, open Computer Management, and start the following components if they are not already started:
  - ReportServer or ReportServer\$InstanceName (application pool)
  - SQL Server Reporting Services (TFSINSTANCE)
- 2. On the application-tier server, open a Command Prompt window, and change directories to Drive:\%programfiles%\ Microsoft Team Foundation Server 11.0\Tools.
- 3. At the command prompt, enter the following command to add the service account for Team Foundation, where DatabaseName is the name of the configuration database (by default, TFS\_Configuration):

### **TfsConfig Accounts /add /AccountType:ApplicationTier** /**account:**AccountName

For more information about how to use this command, see

### Accounts Command.

• Use the **Accounts** command to add the data sources account for the report server and the proxy account for Team Foundation Server Proxy, if your deployment uses these resources.

### **Rebuild the Warehouse**

You can rebuild the data warehouse instead of restoring the TFS\_Warehouse and TFS\_Analysis databases. You will require a significant amount of time to rebuild the warehouse if your deployment contains a lot of data. However, that strategy helps ensure that all data is properly synchronized. When you rebuild the warehouse, Team Foundation Server creates an instance of it, which you must then process to populate it by using data from the operational stores.

#### ⊠Note

If you restored the TFS\_Warehouse and TFS\_Analysis databases in the previous section, you do not have to perform the following procedure.

### To rebuild the warehouse

- 1. On the server that is running the application-tier services for Team Foundation, open a Command Prompt window, and change directories to Drive:\%programfiles%\Microsoft Team Foundation Server 11.0\Tools.
- 2. Enter the following command:

TFSConfig rebuildwarehouse /all /ReportingDataSourcePassword:Password

where Password is the password for the data sources account for Reporting Services (TFSReports).

3. Wait until the command is successfully completed.

#### ⊠Note

If the command is not completed successfully, you should verify that you have all required permissions. For more information, see Troubleshooting the Data Warehouse.

4. On the report server, open Internet Explorer and enter the following string in the Address bar:

http://localhost:8080/VirtualDirectory/TeamFoundation/Administration/v3.0

For VirtualDirectory, enter the virtual directory for Internet Information Services (IIS) that was specified when Team Foundation Server was installed. By default, this directory is named tfs.

The WarehouseControlWebService page opens.

#### ⊠Note

The Microsoft Team Foundation Server Application Pool must be running for the Warehouse Control Web service to be available.

5. Choose GetProcessingStatus, and then choose Invoke.

#### **▲Important**

The service should return a value of Idle for all jobs, which indicates that the cube is not being processed. If a different value is returned, repeat this step until Idle is returned for all jobs.

6. On the WarehouseControlWebService page, choose ProcessAnalysisDatabase, and then choose Invoke.

A browser window opens. The service returns True when it successfully starts to process the cube and False if it is not successful or if the cube is currently being processed.

7. To determine when the cube has been processed, return to the WarehouseControlWebService page, choose GetProcessingStatus, and then choose Invoke.

Processing has completed when the GetProcessingStatus service returns a value of Idle for all jobs.

8. On the application-tier server for Team Foundation, open **Computer Management**, and start the Visual Studio Team Foundation Background Job Service.

### **Clear the Data Cache on Servers**

Each application-tier server in your deployment of Team Foundation uses a file cache so that users can quickly download files from the data-tier server. When you restore a deployment, you should clear this cache on each application-tier server. Otherwise, mismatched file IDs might cause problems when users download files from version control. If your deployment uses Team Foundation Server Proxy, you must also clear the data cache on each server that is configured as a proxy.

#### ⊠Note

By clearing the data caches, you can help prevent the download of incorrect versions of files in version control. You should routinely do this unless you are replacing all hardware in your deployment as part of your restoration. If you are replacing all hardware, you can skip this procedure.

### To clear the data cache

- On a server that is running the application-tier services for Team Foundation or that is configured with Team Foundation Server Proxy, open a Command Prompt window and change directories to Drive:\%programfiles%\Microsoft Team Foundation Server 11.0\Application Tier\Web Services\\_tfs\_data.
- 2. Delete everything in the \_tfs\_data directory.
- 3. Repeat these steps for each application-tier server and each server that is running Team Foundation Server Proxy in your deployment.

### ■Restart Services that Team Foundation Server Uses

After you restore the data, you must restart the services to return the server to an operational state.

### To restart services that Team Foundation Server uses

- 1. On the server that is running the application-tier services for Team Foundation, open a Command Prompt window, and change directories to Drive:\%programfiles%\Microsoft Team Foundation Server 11.0\Tools.
- 2. Enter the following command:

TFSServiceControl unquiesce

For more information, see

TFSServiceControl Command.

### ■Refresh the Caches on Client Computers

### To refresh the cache for tracking work items on client computers

- 1. On the new server, open Internet Explorer.
- 2. In the Address bar, enter the following address to connect to the ClientService web service:

http://PublicURL/VirtualDirectory:8080/WorkItemTracking/v3.0/ClientS

#### ⊠Note

Even if you are logged on with administrative credentials, you might need to start Internet Explorer as an administrator, and you might be prompted for your credentials.

3. Choose StampWorkitemCache, and then choose Invoke.

### ⊠Note

The StampWorkitemCache method returns no data.

### To refresh the version control cache on client computers

- 1. On the client computer, open a Command Prompt window with administrative permissions, and change directories to Drive:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\IDE.
- 2. At the command prompt, enter the following command, including the URL of the collection, which includes the server name and the port number of the new server:

### tf workspaces /collection:http://ServerName:Port/VirtualDirectoryName/CollectionName

In the example deployment, one of Jill's developers needs to refresh the version control cache for a project that is a member of the DefaultCollection collection, which is hosted in the FabrikamPrime deployment of Team Foundation Server. He types the following string:

tf workspaces /collection:http://FabrikamPrime:8080/tfs/DefaultCollection

For more information, see Workspaces Command.

### **See Also**

### Concepts

Team Foundation Server Permissions Team Foundation Server Architecture Restore the Databases Restore Lab Management Components

#### **Other Resources**

Team Foundation Server Services and Service Accounts Restoring a Single-Server Deployment to New Hardware [Tutorial] Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Refresh the Data Caches on Client Computers
See Also Send Feedback

To prevent workspace errors from occurring during version control or build operations in Team Foundation, the data cache on client computers must be updated after certain maintenance operations. After you move, restore, rename, or fail over a data-tier or application-tier server or after you recover from a failure such as a hardware malfunction, you must refresh the cache for tracking work items and users must refresh the version control cache on client computers.

**Required Permissions** 

To invoke the StampWorkitemCache Web method, you must be a member of the Administrators security group on the application-tier server for Team Foundation. For more information, see <u>Team Foundation Server Permissions</u>.

To use the **tf workspaces** command on the client computer, your **Read** permission must be set to **Allow**.

### **Refresh the Work Item Cache**

### ⊠Note

This procedure is optional. You should perform it only if errors occur with work item tracking.

To update the cache for tracking work items, you invoke the StampWorkitemCache Web method. This method forces client computers to update the cache the next time that they connect to the application-tier server. This method also synchronizes the workspaces that are defined on the client computers.

#### ⊠Note

When you invoke the StampWorkitemCache Web method, the performance of Visual Studio Team Foundation Server might temporarily degrade. The performance impact depends on how many Team Foundation users are connected when you invoke the method.

### To refresh the cache for tracking work items on client computers

- 1. On the new server, open Internet Explorer.
- 2. In the Address bar, enter the following address to connect to the ClientService web service:

http://PublicURL/VirtualDirectory:8080/WorkItemTracking/v3.0/ClientS

#### ⊠Note

Even if you are logged on with administrative credentials, you might need to start Internet Explorer as an administrator, and you might be prompted for your credentials. 3. Choose StampWorkitemCache, and then choose Invoke.

### ĭNote

The StampWorkitemCache method returns no data.
# **Refresh the Version Control Cache**

To refresh the version control cache, each user runs the **tf workspaces** command on any computer that must be updated. They must update any computer that uses version control and that connects to a team project collection whose databases were relocated.

### To refresh the version control cache on client computers

- 1. On the client computer, open a Command Prompt window with administrative permissions, and change directories to Drive:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\IDE.
- 2. At the command prompt, enter the following command, including the URL of the collection, which includes the server name and the port number of the new server:

tf workspaces
/collection:http://ServerName:Port/VirtualDirectoryName/CollectionName

In the example deployment, one of Jill's developers needs to refresh the version control cache for a project that is a member of the DefaultCollection collection, which is hosted in the FabrikamPrime deployment of Team Foundation Server. He types the following string:

tf workspaces /collection:http://FabrikamPrime:8080/tfs/DefaultCollection

For more information, see Workspaces Command.

### See Also

#### Tasks

**Open the Team Foundation Administration Console** 

#### **Other Resources**

Workspaces Command Managing Data

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Restore Lab Management Components
See Also Send Feedback

By following the procedures in this topic, you can restore one or more components of Visual Studio Lab Management that have failed. You can decide which procedures to complete based on which components failed, but you should complete the procedures in the following order:

- 1. <u>Restore the library server for System Center Virtual Machine Manager</u> (SCVMM)
- 2. <u>Restore the server that is running SCVMM</u>
- 3. <u>Restore Team Foundation Server</u>
- 4. <u>Restore the Hyper-V hosts</u>

If you follow the recommended order, Team Foundation Server will recognize all virtual machines, templates, hosts and other objects in SCVMM. However, Team Foundation Server will not recognize any objects that were deleted from SCVMM after it was backed up but before Team Foundation Server was backed up. You can remove any such objects from Microsoft Test Manager, as described later in this topic.

Before you can complete these procedures, the following conditions must be true:

- The network names of the new machines must match the names of the machines that were replaced.
- All security accounts must have been restored.
- All software must have been recovered to the same state.

#### **Required Permissions**

To perform this procedure, you must be a member of the following groups:

- The Administrators security group on the server that is running the administration console for Team Foundation.
- The SQL Server System Administrator security group or your SQL Server Perform Back Up and Create Maintenance Plan permissions must be set to Allow on each instance of SQL Server that hosts the databases that you want to back up.
- The Administrator user role in SCVMM.

# **Restore the Library Server for SCVMM**

When you restore the library server, make sure that you use the same names for the host and the library shares. For more information, see the following page on the Microsoft website:

Backing Up and Restoring the SCVMM Database

# **Restore the Server That Is Running SCVMM**

### To restore the server that is running SCVMM

1. Restore the server that is running SCVMM, and give it the same machine name as the old server.

For more information, see the following page on the Microsoft website:

Backing Up and Restoring the SCVMM Database.

• Make sure that the \$machinename of the application-tier server for Team Foundation is a member of the Administrators group on the server that is running SCVMM.

• Add or remove any host groups or library shares that were added or removed after the most recent backup. For more information, see the following page on the Microsoft website: <u>Backing Up and Restoring the SCVMM Database</u>

## **Restore Team Foundation Server**

If Team Foundation Server has failed, you must restore both the application-tier and the data-tier servers that compose the deployment. These components might be hosted on the same server or on multiple servers.

### **To restore Team Foundation Server**

1. Restore the server or servers that are running Team Foundation Server.

For more information, see

<u>Restore an Application-Tier Server</u>, <u>Restore Data to the Same Location</u>, Restore Data to a Different Server or Instance, or Restore a Single Server Deployment to New Hardware.

The machine name, accounts, and software must all be returned to the same state. The machine must be joined and connected to the domain.

If you restore Team Foundation Server to a new machine, you must update the notification URL for Lab Management to match the URL of the new machine. For information about how to set the Lab URL in the Team Foundation Administration Console, see <u>Configuring Lab Management for SCVMM</u> <u>Environments</u>.

• Make sure that Team Foundation Server is running under a user account that is a member of the Administrators group on the server that is running SCVMM. If Team Foundation Server is running as Network Service, the TFS\_MACHINE\$ account should be a member of the Administrators group on the server that is running SCVMM.

• Perform the following steps to make sure that Team Foundation Server is running under a user account that is a member of the Administrators group on all Hyper-V hosts.

1. Open the Team Foundation Administration Console.

- 2. On the Lab Management tab, click Configure Host Groups, and then click Verify.
- 3. When you are prompted for a user name and password, type the user name and password of a member of the Administrators group on all the Hyper-V hosts. The service account for Team Foundation Server is automatically configured for you.
- 4. Repeat steps b and c for each team project collection.

• In SCVMM, manually delete any virtual machines that were created after the point to which you restored Team Foundation Server.

These machines will not appear in either Microsoft Test Manager or Microsoft Environment Viewer. You can find them in SCVMM by looking for the team project name in the description field of the virtual machine.

• In Microsoft Test Manager, manually delete any virtual machines that were deleted after the point to which you restored Team Foundation Server.

These machines are no longer in SCVMM. In both Microsoft Test Manager and Microsoft Environment Viewer, the environment will have a status of Failed, and the virtual machines will have a status as Deleted. If a host was added to a SCVMM host group that was already associated with a team project collection, that host will be available for placement the next time that you deploy an environment.

• In Microsoft Test Manager, manually delete any hosts that were removed from SCVMM after the point to which you restored Team Foundation Server.

Because these hosts can no longer run virtual machines and environments, all environments on these hosts will appear as Failed in Microsoft Test Manager and Microsoft Environment Viewer.

• Manually reassociate any a host groups that were associated with a team project collection and team project after the point to which you restored Team Foundation Server.

The virtual machines on this host group will not be associated with the team project.

• At the levels of both team project collections and team projects, manually unassociate any host groups that were unassociated after the point to which you restored Team Foundation Server.

For more information, see <u>How to: Change the Host Groups for Your Team</u> <u>Project Collections</u>.

• Reassociate any host groups that were associated with a team project after the point to which you restored Team Foundation Server.

• Reassociate any library shares that were associated with a team project after the point to which you restored Team Foundation Server, and reimport all virtual machines and templates that were imported into the team project.

### **Restore the Hyper-V Hosts**

If a host that had virtual environments crashed, they will be in a Failed state.

### To restore a physical host that was running virtual machines

1. Restore the host from your backup, and connect it to SCVMM.

For more information, see the following page on the Microsoft website:

#### Backing Up and Restoring the SCVMM Database.

If the virtual machines are running again on the host, no additional action is required. Lab Management will update the new status of all running virtual machines to the correct state. If you cannot restore the virtual machines and this host is new, the environments will appear as Failed and the virtual machines as Deleted in Microsoft Test Manager or in Microsoft Environment Viewer.

• Refresh Microsoft Test Manager, manually delete the environments that ran on this host, and then re-create them.

If you must determine the environments that ran on this host, SCVMM provides a list of which virtual machines ran on which host.

### See Also

#### Tasks

Move a Team Project Collection

### Concepts

Manually Back Up Team Foundation Server Backing up and Restoring Your Deployment Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Securing Team Foundation Server
See Also Send Feedback

Security for Team Foundation is integrated with Windows authentication and based on users and groups. Windows authentication is used to authenticate accounts for connections between Visual Studio Team Foundation Server and clients of Team Foundation. In addition, authorization in Team Foundation Server is based on users, groups, and permissions in Team Foundation Server. You can increase the security of your deployment in several ways, which include the following tasks:

- Assigning permissions to users and groups.
- Understanding the underlying architecture of your deployment.
- Increasing the security of the network ports and traffic between clients and servers.

You can configure your deployment to use Hypertext Transfer Protocol Secure (HTTPS) with Secure Sockets Layer (SSL). With this configuration, network traffic is encrypted between Team Foundation Server and the clients that request access to its resources, which include team project portals, reports, and work items. You can configure your deployment to support HTTPS with SSL in addition to HTTP, or you can require Team Foundation Server users to use HTTPS with SSL.

## In This Section

Team Foundation Server, Authentication, and Access

Describes the authentication modes Team Foundation Server supports with Visual Studio 2012 installed.

Setting up HTTPS with Secure Sockets Layer (SSL) for Team Foundation Server

Explains the advantages and disadvantages of configuring your deployment to use HTTPS with SSL. This topic also describes how to configure Visual Studio 2012 to support or require HTTPS with SSL for web connections between the data, application, and client tiers of Team Foundation.

### **See Also**

#### Concepts

<u>Team Foundation Server Architecture</u> <u>Team Foundation Server Permissions</u> <u>Configuring Users, Groups, and Permissions</u>

#### **Other Resources**

Administering Team Foundation Server Use Team Web Access to interact with Visual Studio ALM

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Team Foundation Server, Authentication, and Access
See Also Send Feedback

You can configure your deployment to help secure the connections between your users and your deployment of Visual Studio Team Foundation Server. Team Foundation Server (TFS) can support Basic authentication, Digest authentication, and certificates. Therefore, you can configure your deployment of TFS to use Hypertext Transfer Protocol Secure (HTTPS) with Secure Sockets Layer (SSL) and either Basic or Digest authentication. If you adopt this strategy, your users can connect more securely to your deployment without having to use virtual private network (VPN) connections.

# **Configurations**

To support more secure external connections to your deployment of Team Foundation Server, you must configure Internet Information Services (IIS) to enable Basic authentication, Digest authentication, or both. You should also configure any external connections to require certificates.

### **Basic Authentication and Digest Authentication**

Basic authentication is part of the HTTP 1.0 specification and uses Windows user accounts. During Basic authentication, the browser prompts the user for a user name and password and then transmits that information across HTTP using Base64 encoding. By default, Basic authentication requires the Windows user account to have local logon rights at the web server. You can use Basic authentication in both workgroup and domain deployments. Most web servers, proxy servers, and web browsers support Basic authentication, but it is not secure. Because data that is encoded with Base64 is easy to decode, Basic authentication is essentially sending the password as plaintext. By monitoring communications on the network, a malicious user can easily intercept and decipher these passwords by using publicly available tools. To improve security, you should consider using HTTPS with SSL.

Digest authentication is a challenge/response mechanism that sends a digest (also referred to as a hash) instead of a password over the network. During Digest authentication, IIS sends a challenge to the client to create a digest and then send that digest to the server. As a response to the challenge, the client sends a digest that is based on the user's password together with data that is known both to the client and to the server. The server uses the same process as the client to create its own digest, with the user information obtained from Active Directory. IIS authenticates the client only if the digest that the server creates matches the digest that the client creates. You can use Digest authentication only in Active Directory deployments. By itself, Digest authentication is only a small improvement over Basic authentication. A malicious user could record communication between the client and the server and then use that information to replay the transaction. Digest authentication also has dependencies on the HTTP 1.1 protocol, which not all web browsers support.

Moreover, attempts to access Team Foundation Server will fail if you do not configure Digest authentication correctly. Do not choose Digest authentication unless your deployment meets all the requirements for that mode. For more information, see the following page on the Microsoft website (<u>Configure Digest Authentication (IIS 7.0</u>)).

### Authentication Protocols

By default, Team Foundation Server uses the Windows Challenge/Response (NTLM) authentication protocol. NTLM credentials are based on data obtained during the interactive logon process, and include a one-way hash of the password.

Team Foundation Server also supports Microsoft Negotiate as an authentication protocol. In the Negotiate protocol, Kerberos is selected unless it cannot be used by one of the systems involved in the authentication process. For those systems not configured for Kerberos, NTLM is used. Negotiate is the more secure option for most deployments, but can require additional configuration tasks.

### Limitations

In addition to the domain and workgroup requirements that were mentioned earlier in this topic, both Basic and Digest authentication are insufficient by themselves to provide network security for external clients. Therefore, you should not configure Team Foundation Server to support external clients unless you also configure these connections to require HTTPS with SSL.

### See Also

### Concepts

Team Foundation Server Architecture

#### **Other Resources**

Securing Team Foundation Server with HTTPS and Secure Sockets Layer (SSL)

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Setting up HTTPS with Secure Sockets Layer (SSL) for Team Foundation Server

See Also Send Feedback

You can strengthen the security of your deployment of Visual Studio Team Foundation Server (TFS) by configuring the services that it uses to utilize Hypertext Transfer Protocol Secure (HTTPS) with Secure Sockets Layer (SSL). You can configure your deployment either to require this protocol, which maximizes the security of your deployment, or to support HTTPS with SSL in addition to the default protocol, HTTP. Before you choose a configuration, you should carefully review the advantages and disadvantages that this topic describes. After you identify the configuration that best meets the security needs of your organization, you can follow the steps in this topic to configure your deployment.

## In this topic

- Conceptual information
  - Advantages of Supporting HTTPS with SSL in Addition to HTTP
  - Advantages of Requiring HTTPS with SSL for All Connections
  - Disadvantages of Supporting or Requiring HTTPS with SSL
  - <u>Prerequisites</u>
  - <u>Assumptions</u>
- Server configuration
  - Obtaining a Certificate
  - <u>Requesting a Certificate</u>
  - Installing and Assigning the Certificate
  - Configuring Your Firewall
  - Configuring SQL Server Reporting Services
  - <u>Configuring Your Deployment to Support HTTPS with SSL in</u> <u>Addition to HTTP</u>
- Optional configuration tasks
  - <u>Testing Access to Your Deployment (Optional)</u>
  - Configuring Your Deployment to Require HTTPS with SSL (Optional)
- Build configuration
  - Installing the Certificate on Build Servers

- <u>Updating the Build Configuration</u>
- Client configuration
  - 1. <u>Configuring Client Computers</u>

# ■Advantages of Supporting HTTPS with SSL in Addition to HTTP

If you configure your deployment of TFS to support both protocols, users whose computers have been configured for HTTPS with SSL will connect by using that protocol, which makes your deployment more secure. In addition, users whose computers are configured for HTTP only can still connect to your deployment. Although you should not deploy this configuration over public networks, you can gain the following advantages by continuing to support HTTP connections in a controlled network environment:

- You can increase the security of your deployment over time by configuring client computers for HTTPS with SSL as your schedule permits. If you take a phased approach, you do not need to upgrade all computers at the same time, and users whose computers have not yet been upgraded can still connect to the deployment.
- You can more easily configure and maintain Team Foundation Server.
- Calls from one Web service to another are faster over HTTP than over HTTPS with SSL. Therefore, you can continue to support HTTP connections from client computers for which the performance requirements outweigh the security risks.

# **Advantages of Requiring HTTPS** with SSL for All Connections

If you require HTTPS with SSL for all connections, you gain the following advantages:

- All web connections between the application tier, the data tier, and the client tier for Team Foundation are more secure because they require certificates.
- You can control access more easily by configuring certificates to expire when a project phase is expected to end.

# ■Disadvantages of Supporting or Requiring HTTPS with SSL

Before you configure TFS to support or require HTTPS with SSL, you should consider the following disadvantages:

- You might complicate ongoing administration tasks. For example, you might have to reconfigure your deployment to stop supporting HTTPS with SSL before you can apply service packs or other updates.
- You must not only configure but also manage a certification authority (CA) and certificate trusts. You can use Certificate Services in Windows Server 2003 and Windows Server 2008, but you might not want to invest the time and resources that deploying a secure public key infrastructure (PKI) requires.
- You must spend significant time setting up and testing either of these configurations, and troubleshooting your deployment will become more difficult.
- If you continue to support both protocols, external connections might not be encrypted if the application tier for Team Foundation is not appropriately secured.
- If you require HTTPS with SSL, your deployment's performance will be slower.

# ■Configuring Your Deployment to Support or Require HTTPS with SSL

The procedures in this topic describe one process for requesting, issuing, and assigning certificates that are required for SSL connections in Team Foundation Server. If you are using different software than what this topic describes, you might need to perform different steps. To support external connections to your deployment of TFS, you must also enable Basic authentication, Digest authentication, or both in Internet Information Services (IIS).

By following the procedures in this topic, you will accomplish the following tasks:

- 1. Obtain certificates for your deployment of Team Foundation Server and the websites that it uses.
- 2. Install and assign the certificates.
- 3. Configure Team Foundation Server.
- 4. Configure Team Foundation Build.
- 5. Configure client computers.

# **Prerequisites**

To perform the procedures in this topic, you must first meet the following requirements:

- The logical components in the data and application tiers of Team Foundation must be installed and operational. These tiers include IIS, SQL Server, SharePoint Products, Team Foundation Build, and SQL Server Reporting Services if your deployment includes these components. The procedures in this topic refer to the server or servers that are running the logical components in the application tier for Team Foundation as the application-tier server for Team Foundation. The procedures in this topic refer to the server or servers that are running the logical components in the data tier for Team Foundation as the data-tier server for Team Foundation. The application and data tiers might be running on the same server or multiple servers. For more information, see Installing Team Foundation Server.
- You must have a certification authority (CA) from which you can issue certificates. This topic assumes that you are using Certificate Services as your CA, but you can use any CA that you have configured for your deployment. If you do not have a certification authority, you can install Certificate Services and configure one. For more information, see the one of the following sets of documentation on the Microsoft website:
  - For Windows Server 2003:

#### **Certificate Services**

• For Windows Server 2008: <u>Active Directory Certificate Services and Public</u> <u>Key Management</u>

#### **Required Permissions**

To complete these procedures, you must belong to the Team Foundation Administrators group, and you must belong to the Administrators group on the application-tier and data-tier server or servers for Team Foundation. To configure a build server, you must belong to the Administrators group on that server. If your deployment uses SharePoint Products, you must belong to the Administrators group on the server that hosts SharePoint Central Administration. You must also belong to the Farm Administrators group. For more information about permissions, see <u>Team Foundation Server Permissions</u>.

## Assumptions

The procedures in this topic assume that the following conditions are true:

- The data-tier and application-tier server or servers have been installed and deployed in a secure environment and configured according to security best practices.
- You are familiar with how to configure and manage PKIs and requesting, issuing, and assigning certificates. For more information, see the following page on the Microsoft website:

#### Public Key Infrastructure.

• You have a working knowledge of the network topology of the development environment, and you are familiar with configuring network settings, IIS, and SQL Server.

# **Obtaining a Certificate**

Before you configure TFS to use HTTPS with SSL, you must obtain and install a server certificate for the servers in your deployment. To obtain a server certificate, you must install and configure your own certification authority, or you must use a certification authority from an external organization that you trust.

For more information about how to install a certification authority, see the following topics on the Microsoft website:

• For Windows Server 2003:

#### **Certificate Services**

• For Windows Server 2008: <u>Active Directory Certificate Services and Public</u> <u>Key Management</u>

# **Requesting a Certificate**

After you enlist in a certification authority, you must either request a certificate by using IIS Manager, or you must manually install the certificate on each of the following servers in your deployment:

- Each application-tier server.
- Each server that is running Team Foundation Server Proxy, if any are configured for your deployment.
- Each server that is running Team Foundation Build Service as either a build controller or a build agent, if any are configured for your deployment.
- Each server that is running SharePoint Products, if SharePoint Products is configured for your deployment.
- The server that is running Reporting Services, if one is configured for your deployment.

### To request a certificate in IIS

- 1. Choose Start, choose Administrative Tools, and then choose Internet Information Services (IIS) Manager.
- 2. Expand Web Sites or Sites, and navigate to the website for which you want to request a certificate.

For example, to request a certificate for an application-tier server, you navigate to Team Foundation Server. To request a certificate for a proxy server, you navigate to Microsoft Team Foundation Server Proxy.

- 3. Follow the appropriate instructions for your version of IIS to request or create a server certificate that meets the security needs of your organization:
  - If you are using IIS 7.0, prepare a server certificate for IIS 7.0, specify a name for the request, download the certificate, and save it to a secure

location on your server.

For more information, see

Configuring Server Certificates in IIS 7.0.

# Installing and Assigning the Certificate

Before you can use SSL with TFS, you must install the server certificate on the websites that TFS uses, such as the websites in the following list:

- Default Web Site
- Team Foundation Server
- Microsoft Team Foundation Server Proxy
- SharePoint Central Administration

After you install the certificate, you must explicitly bind or assign it to each website, enable the authentication methods that you want to use for each website, and then configure HTTPS for each website.

Depending on your deployment configuration, you might need to install and configure the certificate on more than one computer. For example, your deployment might include Team Foundation Build and SharePoint Products on a different computer from the application-tier server. In this case, you must install and configure the server certificate on not only the server that hosts SharePoint Products but also the computers that host the build controller and build agents.

### **Installing the Server Certificate**

By following these steps, you will install the server certificate or certificates that you want to use for your deployment of TFS.

### To install the server certificate on a website

1. On the server that hosts the website that you want to configure, choose Start, choose Administrative Tools, and then choose Internet Information Services (IIS) Manager.

- 2. Perform the steps appropriate to your version of IIS:
  - If you are using IIS 7.0, import the server certificate from the Server Certificates pane.

For more information, see

Import a Server Certificate (IIS 7).

### ■Enabling Authentication Methods, Enabling HTTPS, and Specifying the Certificate for the Websites That Your Deployment Uses

By following these steps, you can set up the authentication methods that you want to use, and you can enable HTTPS in IIS for the websites that your deployment uses. These websites might be hosted on separate servers. You must perform these steps on each website that you configured in the previous procedure.

After you configure HTTPS, you can additionally secure your deployment by removing HTTP from the list of bindings for each website that you configure.

Depending on your certification hierarchy and public key infrastructure, you should also configure IIS for client certificate authentication. For more information, see <u>Certificate Services</u> and <u>Certificates</u>.

Your deployment of SharePoint Products might require additional configuration, such as alternate access mappings and forms authentication, to operate correctly with HTTPS, SSL, and certificates. For more information, see <u>What Every</u> <u>SharePoint Administrator Needs to Know About Alternate Access Mappings</u>, <u>Forms Authentication in SharePoint Products and Technologies</u>, and the configuration topic for your version of SharePoint Products:

- <u>Configure alternate access mapping (Office SharePoint Server 2007)</u>
- <u>How Do I: Configure an Alternate Access Mapping in SharePoint Server</u> 2010? (video)

### To set up HTTPS and specify the certificate

- 1. On the server that hosts the website that you want to configure, choose Start, choose Administrative Tools, and then choose Internet Information Services (IIS) Manager.
- 2. Perform the set of steps for your version of IIS:

For deployments that use IIS 7.0:

- 1. Expand ComputerName, expand Web Sites, open the submenu for the website that you want to configure, and then choose Edit Bindings.
- 2. In Site Bindings, choose Add.

The Add Site Binding dialog box appears.

3. In the Type list, choose https.

(Optional.) In Port, type a different port number.

#### **▲Important**

The default port number for SSL connections is 443, but you must assign a unique port number for each of the following sites: Default Web Site, Team Foundation Server, Microsoft Team Foundation Server Proxy (if your deployment uses it), and SharePoint Central Administration.

You should record the SSL port number for each website that you configure because you will need to specify these numbers in the administration console for Team Foundation.

In SSL Certificate, choose the certificate that you imported, and then choose OK.

- 4. In Site Bindings, choose Close.
- 5. On the Home page for the website that you are configuring, open the Features view.

- 6. Under IIS, choose Authentication.
- 7. Choose an authentication method that you want to configure, open its submenu, and then choose Enable, Disable, or Edit to enable, disable, or perform additional configuration on the method.

# **Configuring Your Firewall**

You must configure your firewall to allow traffic through the SSL ports that you just specified in IIS. For more information, see the documentation for your firewall.
### ■Configure SQL Server Reporting Services

If your deployment uses reporting, you must configure SQL Server Reporting Services to support HTTPS with SSL and to use the port that you specified in IIS for Team Foundation Server. Otherwise, the report server will not function correctly for your deployment. For more information, see

Configuring a Report Server for Secure Sockets Layer (SSL) Connections. **Tip** 

If your deployment does not use reporting, you can skip this procedure.

# **Configuring Your Deployment**

Follow these steps to configure your deployment with the HTTPS ports and values that you configured in IIS for the default and Team Foundation Server websites.

#### **To reconfigure Team Foundation Server**

1. Open the administration console for Team Foundation.

For more information, see

Open the Team Foundation Administration Console.

- Under Team Foundation, expand the name of the server, and then choose Application Tier.
- In Application Tier Summary, choose Change URLs.

The Change URLs window opens.

• In Notification URL, type the HTTPS URL that you configured for the Team Foundation Server website in IIS.

For example, you might have configured the website to use port 443. In this case, you type https://ServerName:443/tfs. Make sure that you use the fully qualified domain name of the server instead of localhost.

- Choose Test, and then choose OK if the test passes.
- To require HTTPS, choose Use in Server URL, and then type the HTTPS URL that you configured for the Team Foundation Server website.

Make sure that you use the fully qualified domain name of the server instead of localhost.

• Choose Test, and then choose OK if the test passes.

• Open the web.config file for Team Foundation Server in any plain-text editor, such as Notepad. By default, this file is located in the following directory: %PROGRAMFILES%\Microsoft Team Foundation Server 11.0\Application Tier\Message Queue\web.config.

• Remove the following endpoints from the file:

<!-- An empty relative address means the endpoint will pick up the base address of the svc file -->

```
<endpoint address=""
```

```
binding="customBinding"
```

```
bindingConfiguration="TfsSoapBinding"
```

contract="Microsoft.TeamFoundation.Framework.Server.WebServices

/>

<!-- An empty relative address means the endpoint will pick up the base address of the svc file -->

```
<endpoint address=""
```

```
binding="customBinding"
```

bindingConfiguration="TfsSoapBinding"

contract="Microsoft.TeamFoundation.Framework.Server.WebServices

#### />

• Navigate to the <system.serviceModel> section, and add the following protocol mappings:

<protocolMapping>

```
<remove scheme="http" />
```

```
<add scheme="http" binding="customBinding"
bindingConfiguration="TfsSoapBinding"/>
```

<add scheme="https" binding="customBinding" bindingConfiguration="TfsSoapBindingHttps"/>

</protocolMapping>

• In the same <system.serviceModel> section, add the following binding to <customBinding>:

<br/>
<binding name="TfsSoapBindingHttps">

<textMessageEncoding messageVersion="Soap12WSAddressing10" />

<httpsTransport authenticationScheme="IntegratedWindowsAuthentication" manualAddressing="true" />

</binding>

• Save and close the file.

• If your deployment uses SharePoint Products, choose SharePoint Web Applications in the administration console.

• In SharePoint Web Applications, in the Name list, choose a web application, and then choose Change.

The SharePoint Web Application Settings page opens.

• In Web Application URL, change the URL to the HTTPS value for the application.

• In Central Administration URL, change the URL to the HTTPS value for the Central Administration website.

• (Optional) In Friendly Name, change the value to reflect the HTTPS address of this application.

• Choose OK.

• Repeat the previous five steps for every SharePoint web application in your deployment.

• If your deployment uses Reporting Services, in the administration console, choose Reporting.

• In Reporting, choose Edit.

If the Take Offline dialog box opens, choose OK.

The Reporting window opens.

• Choose the Reports tab. In URLs for Report Server, type the HTTPS URLs for Web Service and Report Manager, and then choose OK.

# **Test Access to Your Deployment** (Optional)

You can test whether your changes are functioning as you expect. This step is optional but recommended.

#### To test access to your deployment

- 1. On the server that hosts the application tier, open a web browser.
- 2. In the address bar, type the URL that you use to connect to your deployment through Team Web Access.

#### ⊠Note

You can find this URL on the Application Tier node in the administration console for Team Foundation.

- 3. Verify whether you can access your team project collections and projects from Team Web Access.
- 4. If you cannot access your deployment through Team Web Access, review the steps that you just completed, and make sure that you have made all configuration changes correctly.

### ■Configuring Your Deployment to Require HTTPS with SSL (Optional)

You can require all connections to your deployment to use HTTPS with SSL. This additional security is optional but recommended.

#### **To require SSL connections**

- 1. On the server that hosts the website that you want to configure, choose Start, choose Administrative Tools, and then choose Internet Information Services (IIS) Manager.
- 2. Follow the appropriate steps for your version of IIS:

For deployments that use IIS 7.0:

- 1. Expand ComputerName, expand Web Sites, and then choose the website that you want to configure.
- 2. On the home page for that website, choose SSL Settings.
- 3. In the SSL Settings pane, select the Require SSL check box.

(Optional) Select the Require 128-bit SSL check box.

- 4. In Client Certificates, choose Ignore, Accept, or Require, depending on the security requirements of your deployment.
- 5. In Actions, choose Apply.
- 6. Repeat these steps for each website for which you want to require SSL.

# Installing the Certificate on Build Servers

If you installed Team Foundation Build Service on one or more servers, you must install the certificate in the Trusted Root Certification Authorities store of each server. For more information, see

Obtaining a Certificate and Installing and Assigning the Certificate earlier in this topic. Both the controller and the agent require a certificate with a private key with which to identify themselves in HTTPS connections. **Vote** 

To perform builds over SSL, the certificate must be installed in the trusted root store on both the build controller and the build agent.

# **Updating Build Configurations**

To configure Team Foundation Build for SSL connections, you must configure the build service to use the HTTPS URL that you configured for the application tier and the collection that the build configuration supports. You must configure this URL for each build configuration in your deployment.

#### To change a build configuration to use HTTPS

- 1. On the server that hosts the build configuration that you want to configure, open the administration console for Team Foundation.
- 2. Under Team Foundation, expand the name of the server, and then choose Build Configuration.

The Build Configuration pane appears.

3. Under the service configuration, choose Stop, and then choose Properties.

The Build Service Properties dialog box opens.

- 4. In Communications, make sure that the URL for the team project collection is using the correct HTTPS address and full server name.
- 5. In Local Build Service Endpoint (incoming), choose Change.

The Build Service Endpoint dialog box opens.

- 6. In Endpoint Details, verify that the port number matches your configuration details.
- 7. In Protocol, choose HTTPS.
- 8. In the SSL Certificates list, choose the certificate that you installed and configured for use with this deployment, and then choose OK.
- 9. In the Build Service Properties dialog box, choose Start.

# **Configuring Client Computers**

On every client computer from which users access Team Foundation, you must install the certificate locally and clear the client cache for any user who has accessed Team Foundation from that computer. Otherwise, users will not be able to connect to Team Foundation from that computer. For more information, see

#### Manage Trusted Root Certificates.

#### **▲Important**

Do not follow this procedure for computers that are running both Team Foundation Server and one or more clients of Team Foundation.

#### To install the certificate on a client computer

- 1. Log on to the computer by using an account that belongs to the Administrators group on that computer.
- 2. Install the certificate into the Trusted Root Certification Authorities folder for the local computer.

For more information, see the documentation for your operating system and your certification authority.

#### To clear the cache on a client computer

- 1. Log on to the computer by using the credentials of the user whose cache you want to clear.
- 2. Close any open instances of Visual Studio.
- 3. In a browser window, open the following folder:

#### Drive:\Users\UserName\AppData\Local\Microsoft\Team Foundation\4.0\Cache

- 4. Delete the contents of the Cache directory. Make sure that you delete all subfolders.
- 5. Choose Start, choose Run, type **devenv** /**resetuserdata**, and then choose OK.
- 6. Repeat these steps for the account of every user who has accessed Team Foundation from that computer.

You might want to distribute instructions for clearing the cache to all of your Team Foundation users so that they can clear the caches for themselves.

#### To connect client computers to the reconfigured deployment

• In Visual Studio, connect to Team Foundation Server by using the new HTTPS URL.

For more information, see Connect to Team Projects in Team Foundation Server.

### See Also

#### **Other Resources**

Securing Team Foundation Server with HTTPS and Secure Sockets Layer (SSL) Team Foundation Server, HTTPS, and Secure Sockets Layer (SSL) Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Configuring and Administering Lab Management
See Also Send Feedback

Visual Studio Lab Management lets you create and manage lab environments for a team project on Team Foundation Server. You can use lab environments to develop, test, or run your application. Each machine in a lab environment represents a role required for the application that you intend to develop, test, or run. For example, you might be developing a multi-tiered application that requires three roles: a desktop client, a web server, and a database server. There are two types of lab environments: standard environments and SCVMM environments. A standard environment can contain physical machines and virtual machines using any virtualization platform. An SCVMM environment can only contain virtual machines that are managed by SCVMM on the Hyper-V virtualization platform. For more information about lab environments, see Using a Lab Environment for Your Application Lifecycle.

### **Tasks**

Use the following topics to help you configure and administer Lab Management:

Tasks	Associated Topics
Setting up Lab Management for the first time: To use Lab Management to create standard environments, you need to set up a test controller. To use Lab Management to create SCVMM environments, you must first configure Hyper-V and SCVMM.	• Configuring Lab Management for SCVMM Environments
Changing your Lab Management configuration: You might want to add more host groups, change the library share that is used, modify the integration service account, or add build controllers or	• <u>Changing</u> <u>Existing Lab</u> <u>Management</u> <u>Configurations</u>

test controllers after you have configured Lab Management for the first time.

Assigning users permissions to use Lab Management resources: The default Team Foundation Server security groups automatically give members certain permissions to use Lab Management resources. You might want to modify those permissions or assign permissions to custom groups or users.

<u>Managing</u>
 <u>User</u>
 <u>Permissions</u>
 <u>for Lab</u>
 <u>Management</u>

Backing up and restoring your lab environments: You can back up and restore your lab environments.

- <u>Backing up</u> <u>and Restoring</u> <u>Your</u> <u>Deployment</u>
- <u>Restore Lab</u>
   <u>Management</u>
   <u>Components</u>

### Forums

Visual Studio Lab Management

# Blogs

Visual Studio Lab Management team blog

### See Also

#### Concepts

Using a Lab Environment for Your Application Lifecycle Administering Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Configuring Lab Management for SCVMM Environments
See Also Send Feedback

To create and manage System Center Virtual Machine Manager (SCVMM) environments, you must first configure the required components for Lab Management. These components are:

- Hyper-V hosts
- SCVMM
- Team Foundation Server
- A test controller

For more information about Lab Management, see Using a Lab Environment for Your Application Lifecycle.

#### ⊠Note

Any computers that you use as Library Servers (including the SCVMM machine that has a default library server) and as Hyper-V hosts should be on a gigabit network that is connected to a common network switch. The faster network speed will help improve performance when you deploy virtual machines to hosts and when you save virtual machines to the library from your host.

### **Prerequisites**

The Team Foundation Server administrator must set up and configure Team Foundation Server before configuring Lab Management. For more information about how to set up and configure Team Foundation Server, see the Team Foundation Server Installation Guide. After the administrator has configured Team Foundation Server and created a default team project collection, make sure that you have the following information:

- TfsMachine: The name of the machine that is running Team Foundation Server.
- domain: The Active Directory domain that is used by all machines.
- tfssvc: The account under which Team Foundation Server is running.

#### ⊠Note

You must use a domain user account and password for the TFSservice account to set up an SCVMM environment for lab management.

• domain\tfsadmin: The account for the Team Foundation Server administrator.

### **System requirements**

The following table summarizes the minimum and recommended hardware and software requirements for the machines used with Lab Management. To make the requirements easier to understand, each role is broken out as if it were installed on a separate machine. For a simpler setup, you can run the HyperVHost machine, the VmmMachine and the LibraryMachine all on the same computer.

Role/System	Processor	Supported operating system	RAM (min/ recommended)	Hard disk (min/ recommended)
		Windows Server 2008 R2 operating system with Hyper-V 64- bit, Standard, Enterprise, and Datacenter Editions		
	64-bit with hardware	Windows Server 2008 R2- Server Core installation, Standard, Enterprise, and		100 GB/

HyperVHost	virtualization enabled	Datacenter Editions	4GB / 8GB	200GB
		Windows Server 2008 operating system with Hyper-V 64- bit, Standard, Enterprise, and Datacenter Editions with Service Pack 2		
		Hyper-V Server 2008 R2 operating system		
		Windows Server 2008 R2 operating system 64-bit Standard, Enterprise, and Datacenter Editions		
VmmMachine	64-bit	Windows Server 2008 operating system Standard, Enterprise,	2GB / 4GB	200GB

		and Datacenter Editions with Service Pack 2	
LibraryMachine	64-bit	Windows Server 2008 R2 operating system 64-bit 2GB / 4GB Standard, Enterprise, and Datacenter Editions	200GB / 500GB
TfsMachine	32-bit x86 64-bit x64	All operating systems that are supported by Team Foundation Server are supported by Visual Studio Lab Management, except Windows 8 clients and servers.	120GB / 230GB
		All operating systems that are supported by Team	

		Foundation	
ControllerMachine	32-bit x86 64-bit x64	Server are supported by Visual Studio Lab Management, except any operating systems that are not joined to a domain.	120GB / 230GB

## **Configuration procedures**

The following table details the procedures that you must follow to complete the configuration for Hyper-V, SCVMM, and Lab Management, and the permissions that are required. For a simple setup, the HyperVHost machine, the VmmMachine and the LibraryMachine can be the same machine.

#### ⊠Note

All the machines that are used for Lab Management must be joined either to the same domain or domains that have two-way trust between them.

Step Number	Procedure	Details	Machine Name In Procedure	Permissions Required	Step Complete
1	<u>Set Up And</u> <u>Configure</u> <u>Hyper-V</u> <u>Hosts</u>	Hyper-V lets you manage virtual machines and their resources.	HyperVHost	Administrator for the HyperVHost	
		SCVMM helps you	VmmMachine		
		manage your	⊠Note		
2	<u>Install and</u> <u>configure</u> <u>SCVMM</u>	virtual machines and templates, and where and how	You can use a separate machine as the LibraryMachine to use for the library share to store the	Administrator for VmmMachine	

		you store them.	virtual machines.	
3	Configure Team Foundation Server to enable Lab Management	You configure TFS to give you access to the virtual machines and templates that you have created.	TfsMachine	Team Foundation Server Administrator
4	Verify the configuration using Microsoft Test Manager	Verify that you can connect to a team project and access the Lab Center by using Microsoft Test Manager.	DesktopClient	Any user account that has been added to the computer
	<u>Install test</u>	If you want to build and deploy your application by using Team	TfsMachine <b>☑Note</b> It is recommended	Team Project

5	<u>controller</u>	Foundation Build and run tests, you will need build controllers and test controllers.	that you use a separate machine as the ControllerMachine to install the test controller.	Collection Administrator
		controllerot		

### **Set up and configure Hyper-V hosts**

To set up Hyper-V, you can do one of the following:

• Enable the Hyper-V role on your computer that has Windows Server 2008 R2 installed. If you do not have the Hyper-V role enabled on the HyperVHost computer, you must first enable this by using Server Manager on each of the Hyper-V hosts so that you can create and manage virtual machines using the steps in the following procedure. For more information about Hyper V, see the following

#### Microsoft Web site.

• Install Hyper-V Server 2008 R2 on your HyperVHost. For information about Hyper-V server and how to install it, see the following <u>Microsoft Web site</u>.

#### To enable the Hyper-V role on each Hyper-V host with Windows Server 2008 R2

- 1. On the Hyper-V host, log on as an administrator.
- 2. Choose Start, point to Administrative Tools, and then choose Server Manager.
- 3. In the left pane of Server Manager, right-choose Roles and point to Add Roles.
- 4. Follow the Add Roles Wizard. You can just choose Next in all except the following pages of the wizard:
  - 1. On the Select Server Roles page, select Hyper-V and then choose Next.
  - 2. On the Create Virtual Networks page, you must select an active network adapter and then choose Next.

⊠Note

To find which network adapters are active, use the **ipconfig** command to view which network adapters have IP addresses. In the wizard, select the network adapter that should be used for creating a virtual network. You must select an adapter that is connected to the same network as the other machines in the lab. If there are several adapters, select the adapter to which the virtual machines that were created by Lab Management should be connected.

- 3. On the Confirm Installation Selections page, choose Install.
- 4. On the Installation Results page, choose Close.
- 5. On the Do you want to restart now dialog box, choose Yes.

The machine will restart and continue with Resume Configuration Wizard.

- 6. On the Installation Results page, you will see a message that states **Installation succeeded**. Choose Close.
- 5. Install the update described on this <u>Microsoft Web site</u>. This update supports the network isolation capability of SCVMM environments.

#### **Improve the reliability of WinRM**

Windows Remote Management (WinRM) is used by SCVMM to communicate with Hyper-V hosts. You must make the following changes to the default configuration settings for WinRM. The changes to these settings help improve the reliability of WinRM because Lab Management performs a larger number of operations on Hyper-V hosts than are performed in a typical production scenario for Hyper-V. These changes to the WinRM settings must also be performed on any computer that is used for a library share.

#### **To change Windows Remote Management configuration settings**

1. Choose Start, open the shortcut menu for the Command Prompt, and then choose Run as administrator.

You must be a domain user who has been added to the administrators group for this machine.

2. At the command prompt, run the following command:

#### winrm set winrm/config @{MaxTimeoutms = "1800000"}

3. Restart the winrm service and the SCVMM agent on the host using the following commands:

#### net stop winrm

#### net start winrm

#### net start vmmagent

- 4. If the host OS is Windows Server 2008:
  - 1. Run the following command:

### winrm set winrm/config/Service @{MaxConcurrentOperations="200"}

- 2. Restart winrm and the SCVMM agent on the host using the following commands:
  - net stop winrm

net start winrm

net start vmmagent

#### **Serify the Hyper-V setup**

#### To verify that Hyper-V is installed correctly and is working

• Create a virtual machine using Hyper-V and check that you can start the

virtual machine.

If you have enabled the Hyper-V role for your HyperVHost, create a virtual machine using the information from the following <u>Microsoft Web site</u>.

#### ⊠Note

You can change the settings for the virtual machine based on your needs. You can update the hard disk size and the memory. For more information about these settings, see the following <u>Microsoft Web site</u>.

If you are using Hyper-V Server, create a virtual machine using the information in the Hyper-V Server Getting Started Guide from the following <u>Microsoft Web site</u>.

#### ⊠Note

If you have multiple hard disk drives on your Hyper-V machine, you must configure the paths for the virtual machines to be the larger disk, or you may not have sufficient disk space to create your virtual machines.

For more information about Hyper-V, see <u>this overview of Hyper-V</u>.

#### ⊠Note

When you create or start a virtual machine, issues may occur due to antivirus software. For more information, see the following <u>Microsoft Web site</u>.

# Install and configure SCVMM

To access the virtual machines that you create with Hyper-V from Lab Management, you must install and configure SCVMM. SCVMM is a tool for managing your Hyper-V host machines from a central console. Lab Management communicates with SCVMM to be able to use the virtual machines and templates to create environments. Two versions of System Center Virtual Machine Manager are supported for Lab Management: SCVMM 2012 or SCVMM 2008 R2.

#### ⊠Note

If you are using the stand-alone Hyper-V Server 2008 R2 operating system, you must install SCVMM on a separate VmmMachine. You cannot use the HyperVHost computer. For more information about the unique characteristics of Hyper-V Server 2008 R2, see the following

#### Microsoft Web site.

#### **Install and configure SCVMM 2012**

To install VMM management server, you must have a supported instance of SQL Server and Windows Automated Installation Kit (AIK) for Windows 7 installed. For more information, see this Microsoft website: <u>System Requirements: VMM</u> <u>Management Server</u>

# Install SCVMM server and its administrator console on the VMMMachine

- 1. On VmmMachine, log on as a domain user who is also a local administrator on the machine.
- 2. Run setup.exe.
- 3. In System Center 2012 setup, choose Install.

#### ĭTip

If you do not have Microsoft .NET Framework 3.5 SP1 installed, VMM Setup prompts you to install it. Choose OK.

- 4. In Select features to add, choose VMM management server and then choose Next. VMM Console is automatically installed with the server.
- 5. Type your name, organization and product key and choose Next.
- 6. Choose I have read, understood, and agreed with the terms of the license agreement and choose Next.
- 7. Choose whether to participate in the customer experience improvement program and choose Next.
- 8. To accept the recommended installation location, choose Next.
- 9. Specify the SQL Server instance to use, as well as any particulars about it, and choose Next.
- 10. Specify the local system account or a domain account as the identity the VMM service will use and choose Next.
- 11. Choose Next to accept the default port assignments or to enter different values.
- 12. Choose Create a new library share and choose Next.
- 13. On the Installation summary screen, choose Install, and then choose Close after the wizard finishes.

After you install VMM manager, you must add your Hyper-V host machines to a host group, and optionally add more library shares.

#### **To configure SCVMM**

1. Open Virtual Machine Manager Console.

- 1. Choose Start, All programs, Microsoft System Center 2012, Virtual Machine Manager, and then Virtual Machine Manager Console.
- On the Connect to Server dialog box, type localhost:<port number>or <the name of the SCVMM Server>:<port number>, and then choose Connect.
- 2. To add the HyperVHost machines to a host group in SCVMM, choose Fabric, and then choose Add resources, and then choose Hyper-V Hosts and Clusters. Choose Next in all except the following pages of the wizard:
  - 1. On Specify the credentials to use for discovery, choose Manually enter the credentials, type the credentials of an administrator for the Hyper-V host, and choose Next.
  - 2. On the Specify the search scope for virtual machine host candidates wizard page, choose Specify Windows Server computers by names, in the Computer names box, type the computer name of the host that you are adding to SCVMM (for example, HyperVHost), and then choose Next.
  - 3. On the Select the computer that you want to add as hosts page, select the Hyper-V host computers and choose Next.
  - 4. On the Summary wizard page, choose Finish.

It is recommended that you configure a separate host group for each team project collection. For example, if you plan to configure Lab Management in two project collections in Team Foundation Server, you can create two host groups in System Center Virtual Machine Manager.

3. (Optional) By default, SCVMM configures the machine on which it is installed to act as a library server. It also sets up a default library share on the machine. You can add more library servers or shares to improve the performance.

- 1. To add a new library server to SCVMM, choose Library, and then choose the Add library server button from the ribbon and follow the instructions in the wizard.
- 2. To add a new library share to SCVMM, first create a Windows file share on a library server. Then choose Library, in the navigation pane, select the library server where you created the Windows file share, and choose Library Server from the menu. In the ribbon, choose Add library shares and follow the instruction in the wizard.

We recommend that you configure a separate library share for each team project collection.

- 4. To ensure that virtual machines start quickly and that the host service and any network agents do not timeout, set the minimum amount of memory that the HyperVHost machine will always have in reserve. The default memory reserves on the host machine are often too low to accommodate the longer startup time required for virtual machines.
  - 1. In the SCVMM Administrator Console, choose Fabric.
  - 2. In the navigation pane, expand Servers, All Hosts, choose the Hyper-V host machine, and then choose the Properties button from the ribbon.
  - 3. In the properties dialog box, choose Reserves and then select the Override host reserves settings from the parent host group check box.
  - 4. In the Memory (MB) edit box, type or select 1024, and then choose OK.
  - 5. Repeat steps b. through d. for each host.

#### **Install and configure SCVMM 2008 R2**

#### To install SCVMM server and SCVMM Administrator Console on the VmmMachine

- 1. On VmmMachine, log on as a domain user who is also an administrator on the machine.
- 2. To install System Center Virtual Machine Manager, run setup.exe.
- 3. To install the SCVMM Server, under Setup choose VMM Server. Choose I accept the terms of this agreement. Then choose Next in all except the following pages of the wizard:
  - 1. On the SQL Server Settings wizard page, choose Install SQL Server 2005 Express Edition SP3, and then choose Next.

If you do have an existing instance of SQL that you want to use, then select Use a supported version of SQL Server.

2. On the Library Share Settings wizard page, select Create a new library share.

#### ⊠Note

You can accept the default names and locations for the library share or enter different values.

- 3. On the Installation Settings wizard page, you can either accept the default port numbers and VMM service account or enter different values.
- 4. On the Summary of Settings wizard page, choose Install.
- 5. On the Installation wizard page, choose Close.
- 4. To install the Virtual Machine Manager Administrator Console, under Setup choose VMM Administrator Console. Choose I accept the terms of this agreement. Then choose Next in all except the following pages of the wizard:
- 1. On the Summary of Settings wizard page, choose Install.
- 2. On the Installation wizard page, choose Close.
- 5. On the System Center Virtual Machine Manager 2008 startup page, choose Exit.

After you install VMM manager, you must add your Hyper-V host machine to a host group, and optionally add more library shares.

#### ⊠Note

A host group lets you group the Hyper-V host machines together in a meaningful way based on the purpose of the host machine. For example, a set of host machines might belong to a particular department in your company.

### **To configure SCVMM**

- 1. Open Virtual Machine Manager Administrator Console.
  - 1. Choose Start, All programs, Microsoft System Center, Virtual Machine Manager 2008 R2, and then Virtual Machine Manager Administrator Console.
  - On the Connect to Server dialog box, enter localhost:<port number>or <the name of the SCVMM Server>:<port number>, and then choose Connect.

#### ⊠Note

If you receive an error message that states that you cannot connect because the service is not running, run services.msc and verify that Virtual Machine Manager Service is running.

2. To add the HyperVHost machines to a host group in SCVMM, in the Actions pane choose Add host. Choose Next in all except the following pages of the wizard:

- 1. On Select Host Location wizard page of the Add Hosts wizard, enter the credentials of an administrator for the Hyper-V host.
- 2. On the Select Host Servers wizard page, enter the computer name of the host that you are adding to SCVMM-for example, HyperVHost-and then choose Add.
- 3. On the Summary wizard page, choose Add Hosts.

### ⊠Note

We recommend that you configure a separate host group for each team project collection. For example, if you plan to configure Lab Management in two project collections in Team Foundation Server, you can create two host groups in System Center Virtual Machine Manager.

- 3. (Optional) By default, SCVMM configures the machine on which it is installed to act as a library server. It also sets up a default library share on the machine. You can add more library servers or shares to improve the performance.
  - 1. To add a new library server to SCVMM, choose Add library server in the Actions pane on the right side and follow the instructions in the wizard.
  - 2. To add a new library share to SCVMM, first create a Windows file share on a library server. Then select the library server on the Library tab of SCVMM Administrator Console and choose Add library shares under the Actions pane on the right side.

### ⊠Note

We recommend that you configure a separate library share for each team project collection.

3. On each SCVMM library server, install the update described on this

<u>Microsoft website</u>. This update supports the network isolation capability of SCVMM environments.

- 4. To ensure that virtual machines start quickly and that the host service and any network agents do not timeout, set the minimum amount of memory that the HyperVHost machine will always have in reserve. The default memory reserves on the host machine are often too low to accommodate the longer startup time required for virtual machines.
  - 1. In the SCVMM Administrator Console, choose Go, Hosts.
  - 2. In the center All Hosts pane, open the shortcut menu for the name of the host machine, and then choose Properties.
  - 3. In the Host Group Properties for host machine dialog, choose the Reserves tab.
  - 4. In the Memory edit box, enter or select 1024, and then choose OK. For more information about the minimum required memory for Hyper-V hosts, see this <u>Microsoft website</u>.
  - 5. Repeat steps b through d for each host.

### **▲Important**

If you have two SCVMM servers that are both configured to be used by Lab Management with a specific Team Foundation Server, then each SCVMM server must be configured to use a unique MAC range. To configure the MAC range, from the Virtual Machine Manager Administration Console, in the lowerleft navigation window choose Administration, choose Networking and then open Global Static MAC Address Range. For more information about how to configure the MAC range, see the following <u>Microsoft Web site</u>.

### To verify that SCVMM is installed correctly and is working

1. In the Virtual Machine Manager Administrator Console, in the Actions pane on the right side, choose New virtual machine. Choose Next in all except the following pages of the wizard.

- 1. On the Select Source wizard page, select Create the new virtual machine with a blank virtual hard disk.
- 2. On the Virtual Machine Identity wizard page, in Virtual machine name, enter blank.
- 3. On the Select Destination wizard page, select Place the virtual machine on a host.
- 4. On the Select Host wizard page, view the ratings, and then choose Next.

You should see all the HyperVHost machines that you are using for Lab Management. All of them should have positive star ratings. If not, you might have an issue that must be corrected. To determine why the HyperVHost machines are not usable by SCVMM, examine the Rating Explanation.

5. On the Summary wizard page, choose Create.

A new virtual machine called blank should be created without any errors and should be visible in the Virtual Machines tab of SCVMM. There might be a warning that SCVMM could not locate the boot volume. This is expected and can be ignored.

- 2. In the Virtual Machine Manager Administrator Console, choose Virtual Machines, open the shortcut menu for the virtual machine displayed in the list, and then choose Start. If the virtual machine does not start, review the information located on the following sites to identify the cause of the problem:
  - New Installation of SCVMM 2008
  - System Center Virtual Machine Manager TechCenter
- 3. After the Status of the VM is Running, open the shortcut menu for the VM, and then choose Stop.

This confirms that your Hyper-V hosts are configured correctly.

- 4. To store the virtual machine in the library, open the shortcut menu for the VM, and then choose Store in library.
  - 1. On the Select Library Server wizard page, choose one of the library shares.
  - 2. On the Select Path wizard page, browse to the path for your library share that you plan to use for your team project collection on Team Foundation Server.
  - 3. When you are finished, confirm that a virtual machine called blank appears in the Library pane in the SCVMM Administrator Console.

This confirms that the library share is configured correctly.

You can use the following procedure to improve the performance of virtual machine transfers from the SCVMM library share to a host and also when storing a virtual machine to the SCVMM library share. This performance improvement removes encryption that has security implications if the images are transmitted unencrypted over the network.

### To improve performance of VM transfers

- 1. Enable unencrypted file transfers for library servers.
  - 1. Choose the Library tab in the SCVMM Administrator console. Open the shortcut menu for each library server in the Library Servers tree, and then choose Properties.
  - 2. Select Allow unencrypted file transfers.
  - 3. Choose OK to close the dialog box.
- 2. Enable unencrypted file transfers for host groups.
  - 1. Choose the Hosts tab in the SCVMM Administrator console. Open the shortcut menu for each host group in the Hosts tree, and then choose Properties.
  - 2. Select Allow unencrypted file transfers.

3. Choose OK to close the dialog box.

## Configure Lab Management for Team Foundation Server

To enable Team Foundation Server to communicate with the SCVMM server, you must also install the SCVMM Administrator Console on your TfsMachine if this is a different computer from the computer that you are using as your VmmMachine.

### ĭNote

If your deployment of Team Foundation Server has more than one applicationtier machine, install the console on every application tier in the deployment.

### To install SCVMM 2012 Console on <TFSMachine>

- 1. Logon to TfsMachine as domain\tfsadmin.
- 2. To install Virtual Machine Manager Administrator Console, run setup.exe for Virtual Machine Manager.
- 3. In VMM setup, choose Install.
- 4. You can just choose Next in all except the following pages of the wizard:
  - 1. On the Select features to install page, choose VMM console and then choose Next.
  - 2. On the license page, you must choose I agree with the terms of this notice and then choose Next.
  - 3. On the Installation summary page, choose Install.
  - 4. On the Setup completed successfully page, choose Close.

### To install the SCVMM 2008 R2 Administrator Console on

### <TfsMachine>

- 1. Logon to TfsMachine as domain\tfsadmin.
- 2. To install Virtual Machine Manager Administrator Console, run setup.exe for Virtual Machine Manager.
- 3. Under Setup on the System Center Virtual Machine Manager 2008 R2 startup wizard page, choose VMM Administrator Console.
- 4. You can just choose Next in all except the following pages of the wizard:
  - 1. On the Summary of Settings wizard page, choose Install.
  - 2. On the Installation wizard page, choose Close.

### **Configure Lab Management for Team Foundation Server**

You must configure Lab Management in Team Foundation Server. This enables access to the virtual machines and templates that you create from Microsoft Test Manager. You can then use these virtual machines or templates to create SCVMM environments and store them in a library share. This requires you to select the SCVMM server to use with Team Foundation Server.

### To configure Lab Management for Team Foundation Server

1. On TfsMachine, choose Start, All programs, Microsoft Visual Studio Team Foundation Server 2012, and then choose Team Foundation Server Administration Console.

### ⊠Note

You must log on to TfsMachine as domain\tfsadmin.

- 2. In the Administration Console, in the Application Tier node, choose Lab Management.
- 3. Choose Configure.

The Lab Management Settings dialog box appears.

- 4. On the Virtual Machine Manager tab, in VMM Server Name, type the name of VmmMachine, the server that is running Virtual Machine Manager and that you will use to manage the virtual machines.
- 5. Choose Test to determine whether Team Foundation Server can communicate with the SCVMM server.
  - 1. If Team Foundation Server service account does not have the appropriate permissions on the specified SCVMM, a dialog box prompts you for a user name and password.
  - 2. Specify the credentials of an existing member of SCVMM Administrators Role. By using this user name and password, the Team Foundation Server service account will be added to the SCVMM Administrator Role.

If Team Foundation Server cannot contact the SCVMM server, a red x and an error message is displayed. For more information about how to troubleshoot installation issues, see Troubleshooting Lab Management.

- 6. You can use network isolation in Visual Studio Lab Management to enable multiple copies of a lab environment to run at the same time without causing network conflicts, such as conflicts in computer names and Domain Name System (DNS) registration. For more information about network isolation, see Creating and Using a Network Isolated Environment. On the Network Isolation tab, follow these steps:
  - 1. In IP Block, type the range of Internet Protocol (IP) addresses to be assigned to the virtual machines in an environment when an isolated network is created.

You must type the IP Block using Classless Inter-Domain Routing (CIDR) notation. The CIDR notation is constructed from the IP address, the forward slash (/), and the prefix size. For example, 192.168.0.0/24.

The addresses that you specify are used only for internal routing among virtual machines and are not exposed beyond the boundaries of an environment. Therefore, you can specify any address range that is not used within your public network. In most cases, you can use the default range of 192.168.23.0/24.

2. In DNS Suffix, type the suffix of the domain name to be assigned to the environment when an isolated network is created.

Team Foundation Server uses the suffix that you entered when it registers a unique external name with DNS for each virtual machine in a network-isolated environment. The DNS alias record makes it possible for machines and other objects outside the isolated network to communicate with machines inside the isolated network. Because Team Foundation Server goes into the DNS zone to register the alias record, the service account under which Team Foundation runs must have permissions to add or delete alias records in the specified DNS zone.

If your Team Foundation Server deployment has more than one application tier and each application tier runs under a different service account, then each application-tier service account must have permission to edit the DNS alias records created by the other application tiers.

Choose Test to determine whether Team Foundation Server can use the suffix.

If the suffix is valid, a green check mark appears. Choose OK.

If the suffix is not valid, a red x and an error message appear. You must fix the error before you can continue.

- 7. If your deployment of Team Foundation Server has more than one application-tier machine, repeat the following steps on each of the other application-tier machines:
  - 1. In the Team Foundation Administration Console, choose the Application Tier node, choose Lab Management, and then choose Reconfigure.

The Lab Management Settings dialog box appears.

2. Choose the Virtual Machine Manager tab, and then choose Test to determine whether the application tier can communicate with SCVMM.

If the Team Foundation Server service account does not have the correct permissions on the specified SCVMM, a dialog box is displayed prompting you for a user name and password. Type the credentials of an existing member of SCVMM Administrators role. When you type this user name and password and choose OK, the Team Foundation Server service account of this application tier will be added to the SCVMM Administrator role.

# ■Configure Lab Management for Each Team Project Collection

You must configure the host group and library share from SCVMM that you want to use with each team project collection. For example, you might decide to allocate one host group to each team project collection to evenly divide the virtual machine resources.

To use the workflow capability from a Team Foundation Build build definition to deploy your application, or run tests by using a test controller on your SCVMM environments, you must add a domain user account that will be used by test agents and build agents.

### ⊠Note

You must have a team project collection already created for this step of the configuration.

### To configure Lab Management for each team project collection

1. To select the library share to use to store virtual machines, templates, and SCVMM environments, choose Team Project Collections under Application Tier.

### **▲Important**

You must configure a library share for each team project collection that you want to configure for Lab Management.

- 2. In the right-side pane, choose the appropriate team project collection from the list of project collections.
- 3. Choose the Lab Management tab, and then choose Configure Library

Shares.

The Lab Management Settings dialog box appears.

4. On the Library Shares tab, choose Add and Verify.

The Select Library Shares dialog box appears.

5. In the Select Library Shares dialog box, choose one or more SCVMM library shares that this team project collection will use, and then choose Add.

The verify process now confirms that Team Foundation Server can connect to the library share. If the verification fails, then a log is generated. You can view the log to identify the problems and then fix them. When they are fixed, you can choose Verify to confirm that Team Foundation Server can connect to the library share.

6. To add the library share to each team project in the team project collection, select the Auto Provision check box for the library share.

### ⊠Note

If you select to auto-provision the library share, Team Foundation Server automatically adds the library share to all the team projects in this team project collection. For team projects that have not yet been created, the library share is added when the New Team Project wizard creates the project. For team projects that have already been created in this team project collection, the library share is added when these settings are saved. If you clear Auto Provision the library share will no longer be added to new team projects that you add to your team project collection.

7. To select the host group to use for a team project collection, choose Host Groups.

### **▲Important**

You must select the host group for each team project collection that you

want to configure for Lab Management.

- 8. On the Host Groups tab, choose Add and Verify.
- 9. In the Select Host Groups dialog box, choose one or more SCVMM host groups that this team project collection will use, and then choose Add.
- 10. To add the host group to each team project in the team project collection, choose the Auto Provision check box for the host group.

### ⊠Note

If you select to auto-provision the host group, Team Foundation Server automatically adds the host group to all the team projects in this team project collection. For team projects that have not yet been created, the host group is added when the New Team Project wizard creates the project. For team projects that have already been created in this team project collection, the host group is added when these settings are saved. If you clear Auto Provision, the host group will no longer be added to new team projects that you add to your team project collection.

### Install a Test Controller

You can use Lab Management to deploy an application to your SCVMM environment and also to run tests on that environment. This requires a test controller. For more information about how to run tests using test settings and environments, see Setting Up Test Machines to Run Tests or Collect Data.

### To install the test controller

1. Install the test controller on ControllerMachine or on a different machine. For information about how to install and configure a test controller, see Installing and Configuring Test Agents and Test Controllers.

#### ⊠Note

You must register the test controller with the team project collection that you are using for Lab Management.

2. Specify the user account to use for communication between the test agents and build agents and between the test controllers and Team Foundation Server.

### **A**Caution

The service account that you specify must be a domain account. For security reasons, this account must have limited privileges. The account that you use must not have any administrative permission or be the account that is used by any trusted service. This includes the build controller and test controller.

We recommend that you create two accounts to use for this service account. For more information about how to switch between these two accounts and how existing environments are updated with the service account, see How to: Configure the Lab Service Account.

You must configure the user account for each team project collection that you want to configure for Lab Management.

### See Also

### Concepts

<u>Configuring and Administering Lab Management</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
How to: Create and Store Virtual Machines and Templates Ready for Lab Management
See Also Send Feedback

If you are a system administrator, you might have to regularly create new golden virtual machines to satisfy the needs of various team projects. Because you cannot create a virtual machine (VM) from scratch using Lab Management, you must start from an existing virtual machine created in Hyper-V or System Center Virtual Machine Manager (SCVMM) and import it into a team project. After you create the virtual machines in SCVMM, you can:

- Leave the virtual machine on the host and compose an environment around it.
- Convert the virtual machine into a template and store the template in the team project library.

Do not manually install test agents on your virtual machines if you are going to add those machines to your environment– Lab Management will automatically install test agents on your machines when you add them to a lab environment. However, if you are going to convert a virtual machine into a template, we recommend that you install a test agent on the virtual machine before you convert it into a template. Furthermore, if you are going to use the template to create a network-isolate environment, you must install a test agent on the machine before you convert it in to a template.

The following diagram illustrates how the various machines and agents interact with one another.



This topic contains instructions for creating virtual machines with all the necessary agents by using SCVMM, for converting those virtual machines into templates, and for storing the templates in the SCVMM library.

# ■Prepare a Virtual Machine with a Test Agent

### To create a virtual machine

- 1. Log in as an Administrator to the server that is running Virtual Machine Manager.
- 2. Open the Virtual Machine Manager Administrator Console.
- 3. Create a virtual machine in SCVMM by using one of the following three methods:
  - 1. How to Create a Virtual Machine from an Existing Virtual Hard Disk.
  - 2. <u>How to Create a Virtual Machine with a Blank Virtual Hard Disk</u>.
  - 3. <u>Creating Virtual Machines from a Template</u>.
- 4. Connect to the virtual machine by using the VM console in the SCVMM Administrator Console.
  - 1. In SCVMM, click the Virtual Machines tab, right-click the virtual machine, and then click Connect to virtual machine.
  - 2. Make sure that you have installed the appropriate operating system and application pre-requisites onto this virtual machine.
- 5. Enable remote desktop connections to this virtual machine.

This enables guest-based connections to the virtual machine. For more information about the types of connections, see How to: Connect to machines in a Lab Environment.

1. On the virtual machine, click Start, right-click Computer, and then click Properties.

- 2. In the System dialog box, click Remote settings.
- 3. On the Remote tab of the System Properties dialog box, click the appropriate Allow connections option, and then click Select Users.
- 4. In the Remote Desktop Users dialog box, confirm the appropriate user names appear or click Add to add other user accounts, and then click OK.

You can use Lab Management to deploy an application to your environment and run tests on that environment. This is called workflow integration and testing capabilities for Lab Management. For workflow integration to deploy an application, you create a build definition that includes the workflow to deploy your application to each virtual machine in your environment. You can then run tests on the role that you selected in your test settings. For more information about environments and roles, see Setting Up Test Machines to Run Tests or Collect Data.

Next you will install a test agent on each virtual machine that you created.

### To install a test agent on a virtual machine

• Install a test agent on your virtual machine. For more information about how to install a test agent, see Installing and Configuring Test Agents and Test Controllers.

#### ⊠Note

You do not have to register your test agent with a test controller when you install your test agent. This will be done automatically when the virtual machine is deployed by Lab Management.

If you will compose a virtual environment using this virtual machine, you are finished. If you will store the virtual machine in the library, continue to the next section.

### Caution

Lab Management has very strict requirements for how a network should be configured when you create an isolated environment. Do not do any of the following:

- Set virtual LAN IDs for network adapters. The VLAN ID will be cleared when the virtual machine is deployed. For more information about how to useVLAN IDs, see <u>Configuring Virtual Networks in VMM</u>.
- Create multiple network adapters on a virtual machine that connect to specific networks. Lab Management overrides this configuration and creates two adapters. One adapter connects to the lab network and the other adapter handles communication among virtual machines.
- Configure the MAC address on a network adapter. Lab Management clears the MAC address at the time the network-isolated environment is created.

# ■Store the Virtual Machine into the SCVMM library

When you store a virtual machine in the SCVMM library, you must decide whether to store it as a virtual machine or a template. The decision to store the virtual machine as-is or as a template will depend on several factors:

- Whether applications that are installed on the virtual machine can continue to work if the unique identifying information about the virtual machine is removed.
- Whether virtual machines will be used only as part of a network-isolated environment.

For example, if you will make multiple copies of this virtual machine for use in the same environment, that is, you will use the machine only once in an environment, then compose a virtual environment from the virtual machine without storing it to the library. If you will use the virtual machine in more than one environment on the same network, you must either convert it into a template and store it in library, or configure each environment to use network isolation. For more information about the factors to consider, see Guidance for Creating and Managing SCVMM Environments.

### To store a virtual machine as-is in the library

- Store the virtual machine so that it can be imported into your team project, right-click the VM and point to Store in library.
  - 1. In SCVMM Administrator Console, right-click the virtual machine and then click Shutdown.
  - 2. Select the SCVMM library share that you added to your team project collection and then click Next.
  - 3. Select the path for the share location for this virtual machine and then click Next.

4. Click Store.

# To first convert a VM into a template and then store it in the library

- 1. Prepare the virtual machine to be converted into a template by clearing the administrator password and local password policy.
  - 1. If the virtual machine is joined to a domain, remove it from the domain and join it to a workgroup. Restart the machine when you are prompted to do this.
  - 2. Open gpedit.msc, and then click Local Computer Policy, Computer Configuration, Windows Settings, Security Settings, Account Policies, and then Password Policy.

Set the following values:

Enforce password history: 0

Minimum password length: 0

Password must meet complexity requirements: Disabled

- 3. Close gpedit.msc
- 4. Change the password of local Administrator account to be empty. You can do this by logging in as .\Administrator and then using CTRL+ALT+DEL to change the password.
- 2. Make sure that any applications already installed on the virtual machine will continue to work properly after the name of the virtual machine is changed.

### **▲**Caution

If your application depends on a specific computer name or on joining a specific domain, your application might not work when a new virtual machine is deployed from the template.

- 3. Make sure that there is no media (DVD) loaded into the virtual machine. For more information about how to remove media, see <u>How to Add a DVD</u> <u>or CD Drive to a Virtual Machine</u>.
- 4. Shutdown the virtual machine. In SCVMM Administrator Console, select the virtual machine, right-click it, and then click Shutdown.
- 5. Select the virtual machine and create a clone of it. This is recommended because the original virtual machine is destroyed when it is converted into a template.
- 6. Select the clone, and remove all the checkpoints on the virtual machine. This is required as SCVMM cannot convert a virtual machine with checkpoints into a template.
  - 1. Right-click the virtual machine, and then click Manage checkpoints.
  - 2. Click each checkpoint, and then click Remove.
- 7. If you remove the checkpoints from a virtual machine, Hyper-V merges the disks associated with the checkpoints. This could take several minutes depending on the size of the checkpoints. You can verify the progress of this merging using Hyper-V manager on the host on which the virtual machine is deployed. Wait for the merging to be complete before moving to the next step.
- 8. In SCVMM Administrator Console, right-click the virtual machine, and then click New template.
  - 1. Accept the defaults on all the screens. When prompted for a library share location, select the SCVMM library share.
  - 2. Click Create to start the SCVMM job. If the SCVMM job is not completed successfully, review the error message and take the appropriate actions.

### See Also

### Concepts

<u>Configuring Lab Management for SCVMM Environments</u> Creating Lab Environments Creating and Using a Network Isolated Environment

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
How to: Prepare a Domain Controller Virtual Machine Using SCVMM
See Also Send Feedback

A virtual environment that uses network isolation requires that the virtual machines in the environment must be either in a workgroup or joined to a private domain that is served by a domain controller (DC) virtual machine within the environment. Among the steps to set up an environment that uses network isolation is to prepare a domain controller virtual machine. For more information about the other steps, see Creating and Using a Network Isolated Environment.

**Required Permissions** 

To perform these procedures, you must be a member of the following groups:

• Either the Administrator user role or the Self-Service User role in System Center Virtual Machine Manager (SCVMM).

For more information about SCVMM roles, see <u>Managing User Roles</u>.

### To prepare a domain controller virtual machine

- 1. Use SCVMM Administrator Console to create a new virtual machine with the Windows Server 2008 R2 operating system.
  - 1. We recommend that you use a new installation of the operating system instead of reusing or customizing an existing image.
  - 2. Make sure that you allocate at least 1GB of memory your virtual machine.
- 2. Install Active Directory and DNS server roles in the virtual machine. For more information about how to install Active Directory on Windows Server

2008 R2, see AD DS Installation and Removal Step-by-Step Guide.

- 1. Lab Management does not support use of separate virtual machines in an environment for Active Directory and DNS.
- 2. Make sure that the domain controller is the only DC in the forest and that it is the root of the forest.
- 3. Do not use a domain name that is already used in your corporate or lab network. For example, if one of the domains in your network is called test.contoso.com, do not use the same name for your private domain.
- 3. Install the test agent on the domain controller virtual machine. For more information, see Installing and Configuring Test Agents and Test Controllers.
- 4. Verify that the Active Directory and DNS are installed correctly.
- 5. If the virtual machine was ever connected to a network, make sure that the only DNS entries are for the virtual machine. You only have to perform these steps one time: after you install Active Directory and DNS and before you store the virtual machine into library. Perform these steps and then immediately shut down the virtual machine. If you restart the virtual machine after you perform these steps, or if you leave the virtual machine running for a long time after you perform these steps, you must repeat them.
  - 1. Open the DNS tool by clicking Start, Administrative Tools, and then DNS.
  - 2. In the DNS tool, navigate to Forward Lookup Zones. Select each zone that is listed under Forward Lookup Zones. Then delete all the DNS entries under that zone that have an IP address listed in the Data column.
  - 3. In the DNS tool, navigate to Reverse Lookup Zones. If any reverse lookup zone is configured, delete all the DNS entries under that zone.
  - 4. Close the DNS tool.

- 6. Shut down the virtual machine, and store it in the SCVMM library.
  - 1. Do not turn off the Active Directory VM. You have to shut it down correctly.
  - 2. Do not generalize the Active Directory VM, either by running Sysprep or by storing the virtual machine as a template in SCVMM.

### See Also

### Concepts

Creating and Using a Network Isolated Environment Configuring Lab Management for SCVMM Environments

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Changing Existing Lab Management Configurations
See Also Send Feedback

After you have configured Lab Management, you might want to change the resources that you are using for a team project collection. You might want to add more resources as you create more virtual machines, templates, and environments, or change the integration service account, or add a build controller or test controller.

### **Tasks**

Use the following topics to help you change existing Lab Management configurations:

Tasks	Associated Topics
Change the host groups for your team project collection: You can add or remove host groups from your team project collection by using the Administration Console for Team Foundation, or you can use the command-line utility TFSLabConfig.	• <u>How to:</u> <u>Change the</u> <u>Host Groups</u> <u>for Your</u> <u>Team Project</u> <u>Collections</u>
Change the library shares for your team project collection: You might want to add another library share to provide more storage for your virtual environments, templates, or virtual	• <u>How to:</u> <u>Change the</u> <u>Library Share</u> <u>for Your</u> <u>Team Project</u> <u>Collections</u>

### machines.

Change the integration service account that provides communication between the agents and controllers: We recommend that you change this user account regularly for security reasons.

 Setting Up Test Controllers in Lab Environments

Change Active Directory domain to which the SCVMM server belongs. • <u>How to: Join</u> <u>the SCVMM</u> <u>Server to</u> <u>another</u> <u>Domain</u>

### See Also

### Concepts

Configuring Lab Management for SCVMM Environments Configuring and Administering Lab Management Administering Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
How to: Change the Host

How to: Change the Host Groups for Your Team Project Collections See Also Send Feedback

You can use the host groups in System Center Virtual Machine Manager as a convenient way to monitor and manage virtual machine hosts. Hosts are the computers in the host group that you use to deploy virtual environments in Visual Studio Lab Management. You create and manage host groups by using SCVMM. In Lab Management, you can specify which host groups to use for a team project collection.

If you want to add a host to an existing host group, you must use SCVMM to accomplish this task. For more information, see this <u>Microsoft Web site</u>. If you do add a host, you must also verify the following:

- The logical network of the host group is the same as the other hosts so that the virtual machines can communicate with each other.
- The host has been added to the correct host group.

After you have configured Lab Management for the first time, you might later want to change the host groups that are used by your team project collection. For example, you need more hosts because the volume of use has increased. You can choose to add another host group that you can use to your existing team project collection to increase your capacity. You can add multiple host groups to any team project collection.

### ⊠Note

To configure Lab Management for the first time, see <u>Configuring Lab</u> <u>Management for SCVMM Environments</u>.

You can change the host groups for a team project collection using the Administration Console for Team Foundation. Or, you can use the command-line to accomplish the same task. For more information about the command-line utility TFSLabConfig, see <u>Configuring Lab Management with TFSLabConfig</u>.

### ⊠Note

If you want to assign a host group that you have created using SCVMM to a specific team project that you have added to your team project collection, you must use the command-line utility to accomplish this task. You cannot use the administration console to assign host groups to specific team projects. For more information, see <u>TFSLabConfig CreateTeamProjectHostGroup Command</u>.

Use the following procedure to change the host groups for your team project collection by using the Administration Console for Team Foundation. For each host in a host group that you add, you must also run the steps in the procedure to improve the reliability of WinRM. To expand your virtual lab, see <u>Expanding or Changing Your Virtual Lab</u>.

### To add or remove host groups for your team project collection

- 1. In the Team Foundation Administration Console, under Application Tier, click Team Project Collections.
- 2. In the right-side pane, click the appropriate team project collection from the list of project collections.
- 3. Click the Lab Management tab, and then click Configure Host Groups.

The Edit Project Collection Level Lab Management Settings dialog box is displayed.

- 4. To add a host group, on the Host Groups tab, click Add and Verify.
  - 1. In the Select Host Groups dialog box, click one or more SCVMM host groups that this team project collection will use, and then click Add.
  - 2. To add the host group to each team project in the team project
collection, select Auto Provision for the host group.

#### ⊠Note

If you select to auto provision the host group, Team Foundation Server automatically adds the host group to all the team projects in this team project collection. For team projects that have not yet been created, the host group is added when the New Team Project wizard creates the project. For team projects that have already been created in this team project collection, the host group is added when these settings are saved. If you clear Auto Provision the host group will no longer be added to new team projects that you add to your team project collection.

5. To delete a host group from your team project collection, on the Host Groups tab, select the host group from the list, and click Delete.

#### **▲Important**

Before you can delete a host group from a team project collection, you must first remove all Lab Management environments that are in this host group using Microsoft Test Manager. Then you must remove the association of this host group from each team project in that project collection using the command line utility TFSLabConfig. For more information about how to do this, see <u>TFSLabConfig</u> <u>DeleteTeamProjectHostGroup Command</u>.

6. Click OK.

# **Improve the Reliability of WinRM**

You must make following changes to the default configuration settings for Windows Remote Management for each host in a host group that you have added. The changes to these settings help improve the reliability of WinRM because Lab Management performs a larger number of operations on Hyper-V hosts than are performed in a typical production scenario for Hyper-V.

### To change Windows Remote Management configuration settings

1. Click Start, right-click Command Prompt, and then click Run as administrator.

#### ⊠Note

You must be a domain user that has been added to the administrators group for this machine.

2. At the command prompt, run the following command:

#### winrm set winrm/config @{MaxTimeoutms = "1800000"}

3. If the host operating system is Windows Server 2008 SP2, also run the following command:

## winrm set winrm/config/Service @{MaxConcurrentOperations="200"} -r:http://<HyperVHost>

(Replace HyperVHost with the computer name of the Hyper-V host.)

#### ⊠Note

To create a WinRM listener on http://<HyperVHost>, you might also have to run the following command:

#### winrm quickconfig

4. Restart winrm and the SCVMM agent on the hosts after you make these changes using the following commands:

net stop winrm

net start winrm

net start vmmagent

## See Also

#### Tasks

Create a Team Project Collection

### Concepts

Configuring Your Server Using the Team Foundation Administration Console

#### **Other Resources**

**Changing Existing Lab Management Configurations** 

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
How to: Change the Library Share for Your Team Project Collections
See Also Send Feedback

You can use the library shares in the System Center Virtual Machine Manager (SCVMM) library as a place to create and store the virtual machines, templates, and environments in your team projects. You use SCVMM to create and manage the library shares and to specify on which computer a library share is located.

After you have configured Lab Management for the first time, you might later want to change the library shares that are used by your team project collection.

#### ⊠Note

To configure Lab Management for the first time, see <u>Configuring Lab</u> <u>Management for SCVMM Environments</u>.

Using SCVMM, you can create multiple library shares on a single library server. The library shares enable you to allocate the disk space on your library server. If you do this you can allocate disk space for each team project collection. You can add multiple library shares to each team project collection. For example, you can add an extra library share from a different library server to a specific team project collection to provide more disk space for the environments, virtual machines, and templates for the teams that use that team project collection. If you are maximizing the network bandwidth to a specific library server for your team project collection, you may want to add a library share from another library server. Or, you might want to add a library share that has been added to a library server in a different physical location.

#### ⊠Note

For improved performance, it is better to locate library shares and host groups

in the same physical location. For example, if you have a distributed team with half your team located in one country and the other half in another country, then it is more efficient to have library shares and host groups that are physically located in each of those countries and have the team members use the library shares that are located together in their country. You can change the library shares by using the Administration Console for Team Foundation. Or, you can use the command-line to accomplish the same task. For more information about the command-line utility TFSLabConfig, see <u>Configuring Lab Management</u> with TFSLabConfig.

Use the following procedure to change the library shares for your team project collection using the Administration Console for Team Foundation. For each library share that you add that is located on a different library server, you must also run the steps in the procedure to improve the reliability of WinRM. To expand your virtual lab, see <u>Expanding or Changing Your Virtual Lab</u>.

## To add or remove library shares for a team project collection

- 1. In the Team Foundation Administration Console, under Application Tier, click Team Project Collections.
- 2. In the right-side pane, click the appropriate team project collection from the list of project collections.
- 3. Click the Lab Management tab, and then click Configure Library Shares.

#### ⊠Note

You must create a library share using SCVMM before you can add this to be used by your team project collection.

4. To add a library share, on the Library Shares tab, click Add and Verify.

The Select Library Shares dialog box is displayed.

1. Click one or more SCVMM library shares that this team project collection will use, and then click Add.

2. To add the library share to each team project in the team project collection, select Auto Provision for the library share.

#### ⊠Note

If you select to auto provision the library share, Team Foundation Server automatically adds the library share to all the team projects in this team project collection. For team projects that have not yet been created, the library share is added when the New Team Project wizard creates the project. For team projects that have already been created in this team project collection, the library share is added when these settings are saved. If you clear Auto Provision the library share will no longer be added to new team projects that you add to your team project collection.

5. To delete a library share, on the Library Shares tab, select the library share from the list, and then click Delete.

#### **▲Important**

Before you can delete a library share from a team project collection, you must first remove all Lab Management environments, virtual machines and templates that are stored on this library share using Microsoft Test Manager. Then you must remove the association of this library share from each team project in that project collection by using the command-line utility TFSLabConfig. For more information about how to do this, see <u>TFSLabConfig DeleteTeamProjectLibraryShare Command</u>.

- 6. To confirm that Visual Studio Team Foundation Server can connect to the library share, click Verify.
- 7. Click OK.

# **Improve the Reliability of WinRM**

You must make following changes to the default configuration settings for Windows Remote Management for each library server that has library shares that you use for Lab Management. The changes to these settings help improve the reliability of WinRM because Lab Management performs a larger number of operations on Hyper-V hosts than are performed in a typical production scenario for Hyper-V.

### To change Windows Remote Management configuration settings

1. Click Start, right-click Command Prompt, and then click Run as administrator.

#### ⊠Note

You must be a domain user who has been added to the administrators group for this machine.

2. At the command prompt, run the following command:

#### winrm set winrm/config @{MaxTimeoutms = "1800000"}

3. If the host operating system is Windows Server 2008 SP2, also run the following command:

## winrm set winrm/config/Service @{MaxConcurrentOperations="200"} -r:http://<LibraryServerMachine>

(Replace LibraryServerMachine with the computer name of the library server.)

#### ⊠Note

To create a WinRM listener on http://<LibrayServerMachine>, you might also have to run the following command:

### winrm quickconfig

4. Restart winrm and the SCVMM agent on the hosts after you make these changes using the following commands:

net stop winrm

net start winrm

net start vmmagent

## See Also

#### Tasks

Create a Team Project Collection

### Concepts

Configuring Your Server Using the Team Foundation Administration Console

#### **Other Resources**

**Changing Existing Lab Management Configurations** 

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
How to: Configure the Lab Service Account Send Feedback

This topic has been superseded by Setting Up Test Controllers in Lab Environments.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Expanding or Changing Your Virtual Lab
See Also Send Feedback

As you use and become familiar with a virtual lab, you might want to either expand or change your lab in one or more ways:

Move Team Foundation Server from One Server to Another

You might have to change or replace the server that is running the Team Foundation Server application-tier.

<u>Move a Team Project Collection from One Team Foundation Server to</u>
 <u>Another</u>

You might want to reuse the machines and other resources in your initial team project and move those resources.

• Add a New Library Server and Retire an Old Library Server

You might find you need more space to store virtual machines, templates, and environments. You can add more library shares or replace existing library shares with ones on larger SCVMM library servers.

• Add a New Hyper-V Host and Retire an Old Hyper-V Host

You might find you need more space to run virtual machines and environments. You can add more physical hosts to existing host groups to transparently add more capacity.

 Move System Center Virtual Machine Manager from One Server to Another

You might have to change or replace the server that is running SCVMM.

• <u>Move a Test Controller</u>

You might have to move or replace the machine that is running the test controller.

Each of these changes requires some modification to the configuration of Lab Management resources.

## ■Move Team Foundation Server from One Server to Another

If you move Team Foundation Server from one server to another, or otherwise change the name of the server that is running Team Foundation Server, you must update the URL that the Lab Management components use to communicate with Team Foundation Server. For more information about how to move Team Foundation Server, see

#### Administering Team Foundation Server.

After you have finished moving Team Foundation Server, update the Team Foundation Server URL in the following locations:

- On the machine that is running the Team Foundation Server applicationtier, the URL appears in the Team Foundation Server Administration Console. In the Administration Console, go to the Lab Management node, click Reconfigure Lab Management, and then click the Advanced tab. In the Lab URL text box, type the new server name.
- On machines where a build agent is installed, the URL is stored in the Hyper-V-managed registry hive. If the server that is running Team Foundation Server is renamed, click Repair Workflow Capability for this environment. The setting will be updated automatically in all the virtual machines of this environment. Because .lvr files store the URL of Team Foundation Server internally, you must modify the .lvr file manually to update the XML.

#### ⊠Note

You do not have to revert snapshots, because the URL is updated in the virtual machines by Team Foundation Server after it reverts to a snapshot.

## ■Move a Team Project Collection from One Team Foundation Server to Another

If you started to explore Lab Management by using composed environments or by creating a small team project to pilot test Team Foundation Server, you might want to migrate the lab assets to a different Team Foundation Server instance later. Because Team Foundation Server does not support merging one team project with another, you must move your lab resources manually. For more information about how to migrate source code, work items, and build definitions, see

#### Administering Team Foundation Server.

#### ⊠Note

You can move virtual machines only within the same network location because they are already configured to use that network location. If you move a team project collection between instances of Team Foundation Server that are located in different domains, you must move the project collection, un-configure Lab Management for that project collection, and then configure it from fresh using the settings of the new Team Foundation Server instance and domain.

Because you are moving between domains, you cannot recover everything.

## To move from one server that is running Team Foundation Server to another server that is running Team Foundation Server

- 1. Move the team project collection to a new application tier. For the steps see <u>Move a Team Project Collection</u>.
- 2. Un-configure Lab Management for the team project collection after attaching the collection to the destination Team Foundation Server. For the steps to un-configure, see <u>TFSConfig Lab /Delete Command</u>.

3. Reconfigure Lab Management for the collection in the destination Team Foundation Server. For the steps to configure Lab Management, see <u>Configuring Lab Management for SCVMM Environments</u>.

## ■Add a New Library Server and Retire an Old Library Server

If you must expand the storage for System Center Virtual Machine Manager (SCVMM), you can add new library shares. You can also replace existing shares with shares on servers that have faster processors, more hard disks, or both. The process to add or remove library shares to the SCVMM used by Lab Management is as follows:

- 1. Prepare to move to a new library share.
- 2. Move the templates.
- 3. Move the imported virtual machines that are currently not used in virtual environments.
- 4. Move the stored environments.
- 5. Remove the old library share.

#### ⊠Note

This scenario will not update existing .lvr files that may be part of active bugs. If there are stored environments that are required for reproducing bugs, those should remain in the source library until the bugs are resolved.

### To prepare for the move

1. Make sure that the new library server meets the system requirements for a SCVMM library server. For more information about the system requirements, see

#### System Requirements: VMM Library Server.

• On the new server, add a new library share:

- 1. In Windows Explorer, create a new folder that uses the name of the new library share.
- 2. Share the new folder with the SCVMM server by adding VMM\_MACHINE\$ as a contributor to the list of users this folder is shared with.
- 3. In the Virtual Machine Manager Administrators Console, associate this library share.

• Add the new library share to the relevant team project collections and team projects.

- 1. In the TFS Administration Console, click TFS Server. Click Application Tier, and then click Team Project Collections.
- 2. On the Lab Management tab, click Configure library shares.
- 3. Add the share that you create in SCVMM.

• Back up the original library share. For the steps to back up the library share, see Backing Up and Restoring the VMM Database.

• Do one of the following:

Take the project collections offline. It is recommended, although not required, to take the relevant project collections offline during the move.

-or-

Contact your system administrator and ask them to notify all users not to deploy environments or import virtual machines or templates during the time of the move.

## To move templates

1. Identify the files that are related to the template that you want to move. Move those files from the source to the destination. In the case of differencing-disk VHDs, make sure that you move all the versions of the .vhd. To find the path of the .vhd that this template points to, right-click the template and then click Properties. Click Hardware Configuration. Move this .vhd to the new library share.

- 2. In the SCVMM Administrators Console, click Library. In the Action pane, click Refresh. Refresh both the old library share and the new library share.
- 3. In Microsoft Test Manager, click Library. Point to the new Library Share, and re-import the templates. Repeat for each team project that is enabled for Lab Management. Make sure that the OS profile and the hardware profile that you define to the new instance of the template are the same as the one that you used for the old template.
- 4. If you have environments in the library that are based on templates, you should update the environment definitions manually. In Microsoft Test Manager, click Library. Click Environments and then click each environment that used the templates. Remove the old template and add the new template.
- 5. Delete the old templates from the library. In Microsoft Test Manager, click the Lab Center. Click the Library tab, and then click VMs and Templates.

You can distinguish between the old and the new templates by looking at the library share column. This does not delete the underlying template.

6. Confirm the move was successful. If you did not take the team project collection offline, deploy one the templates to a host. It should complete without errors.

If there are virtual machines in a library that are currently not used in an environment, you must deploy those virtual machines to a host and then save those virtual machines to the new library.

### To move imported virtual machines

- 1. Create a new virtual environment
- 2. Add all the virtual machines that must be moved. Do not select any capabilities.
- 3. Deploy the environment, but do not start it.

#### ⊠Note

If you try to start the environment, and there are running virtual machines that are using the same name of the virtual machines you are now moving, name conflicts in the domain might occur.

- 4. In Microsoft Test Manager, click the first virtual machine in the environment. Click Store to Library and then select the target library you want to store to. Repeat for all the virtual machines in the environment.
- 5. After all the virtual machines are stored in the library, delete the environment
- 6. Delete the virtual machines in the source library.

#### ⊠Note

After the move, you will not be able to import the virtual machines into other team projects.

#### To move stored environments

1. In Microsoft Test Manager, deploy the environment.

You do not have to start the environment.

- 2. On the Environment tab, click Store to library on the destination library share.
- 3. Confirm the move was successful:
  - 1. Deploy the environment.
  - 2. Take a snapshot.
  - 3. Revert to an earlier snapshot.
  - 4. Revert to the latest snapshot.

4. If you can successfully revert, delete the environment from the source library share.

After the move is complete, place the team project collection back online.

If you have bugs that included snapshots for environments that are in the library, we recommend that you keep the environments in this library until those bugs are resolved. There is no easy way to find which environments has .lvr files pointing to them. Therefore, a practical solution will be to keep environments that were created in the last 60 days.

## To remove the share

1. Delete virtual machines and templates from the original library share: In Microsoft Test Manager, click Lab Center. Click Library, Virtual Machines and Templates and then delete the virtual machines in the library share that are still shown at located in the original library share.

You can easily group the virtual machines by library share, by dragging the Library Share column header and dropping it above the header row.

- 2. Deleting library share association from team projects:
  - 1. Open a command prompt.

At the command prompt, use the TFSLabConfig command line tool for each project that is associated with the library share that you want to delete.

2. Run the command **TFSLabConfig DeleteTeamProjectLibraryShare**, to specify the project collection and project name, in addition to the library share name.

For more information about the **TFSLabConfig DeleteTeamProjectLibraryShare** command, see <u>TFSLabConfig</u> <u>DeleteTeamProjectLibraryShare Command</u>.

- 3. Repeat this step for each team project.
- 3. In the Team Foundation Server Administration Console, delete the original

library share from the team project collections that it was associated with.

## ■Add a New Hyper-V Host and Retire an Old Hyper-V Host

## To add a Hyper-V host

1. In the SCVMM Administrators Console, add the new Hyper-V server to the host group. For the steps to add a server to the host group, see

#### Adding Hosts.

• In the Team Foundation Server Administration Console, click TFS Server. Click Lab Management and then click Host Groups. Click Verify. If you are prompted for a username and password, provide a user who is a member of the Administrators security group on the new host.

• Prevent placement of new virtual machines on the host that you are migrating from.

- 1. In the SCVMM Administrators Console, click Hosts. Click the name of the host that you are migrating from, and then click Properties.
- 2. In the Host Properties window, click the Status tab. Clear the check box This host is available for placement, and then click OK.

• In the Host Properties window, click the Status tab. Clear the check box labeled This host is available for placement and then click OK.

- Migrate environments that are not network-isolated:
  - 1. In Microsoft Test Manager, click Lab Center. Click the Lab tab and then click Environments. Identify the environments that reside in the host group in which the source and destination hosts are.
  - 2. Open each environment and identify which of them have the Network Isolation capability off. The rest of this step applies only to those environments that are not network-isolated.

3. In SCVMM Administrators Console, find the virtual machines that are part of the environments that are not network-isolated.

You can do this by typing the environment name in the search box of the Virtual Machines pane.

- 4. If any of the virtual machines that were found in the previous step reside on the host that is being discontinued, move those virtual machines to the new host.
- Migrate environments that are network-isolated:
  - 1. In Microsoft Test Manager, click Lab Center. Next, click the Lab tab and then click Environments. Identify the environments that reside in the host group where the source and destination hosts are.
  - 2. Open each environment and identify which of them have the Network isolation capability on.
  - 3. In SCVMM Administrators Console, find the virtual machines that are part of network-isolated environments.

You can do this by typing the environment name in the search box of the Virtual Machines pane.

4. If the virtual machines of any network-isolated environment reside on the host being discontinued, store the network-isolated environment in the team project library.

All virtual machines within a network-isolated environment are located on the same physical host.

- 5. Deploy the environment from the library back to the host group.
- In Microsoft Test Manager, pause and then resume each environment. This verifies that the environment was successfully migrated.

• In the SCVMM Administrators Console, click Hosts, click the name of the old host, and then click Remove host.

## ■Move System Center Virtual Machine Manager from One Server to Another

There are at least two situations in which you might move SCVMM from one server to another. First, you began to use Lab Management on a pilot basis on temporary machines and now want to deploy the functionality to production servers. Because you are starting over in a production mode, it is not important to retain the environments, templates, and other lab assets you had been using during pilot testing. Second, you have to change the server that is running SCVMM because of capacity or availability issues. Because you want to continue operating as before, you have to preserve the same environments and related assets.

## To move SCVMM without retaining lab assets

- 1. Delete the lab objects from each team project collection.
  - On the Team Foundation Server application tier, open a command prompt and type the following command Tfsconfig.exe lab /delete /collectionName:myCollection. You must run TFSConfig from a server that is running the Team Foundation application tier. By default, TFSConfig is located in Drive:\Program Files\Microsoft Team Foundation Server 2010\Tools. For more information about the TfsConfig Lab /Delete command, see

### TFSConfig Lab /Delete Command.

- Repeat this command for each team project collection on the server.
- Enter the new name in Team Foundation Server Administration Console.
  - 1. In Team Foundation Server Administration Console, click the Lab Management node.

- 2. On the Lab Management page, click Reconfigure Lab Management.
- 3. On the Virtual Machine Manager tab, type the fully-qualified domain name of the new server that is running SCVMM. Then click Test to confirm that the new server is available.
  - 1. If the Team Foundation Server service account does not have the appropriate permissions on the specified SCVMM, a dialog box prompts you for a user name and a password.
  - 2. Type the name and the password of a user who is an existing member of the SCVMM Administrator role. The Team Foundation Server service account will be added to the SCVMM Administrator role.
- 4. Click OK.

### To move SCVMM while retaining lab assets

- 1. Back up the database of the original SCVMM:
  - 1. In the SCVMM Administrators Console, click Administration and then click General. In the Actions pane, click Back up Virtual Machine Manager.
  - 2. In the Virtual Machine Manager Backup dialog box, type the path of a destination folder for the backup file. The folder must not be a root directory and must be accessed by SQL Server.

#### ⊠Note

You can follow the status of the backup in Jobs view.

- 2. Copy the database backup files from step 1b to a location on the new SCVMM machine.
- 3. Install SCVMM on the new server. Either point the wizard to the current SCVMM database or create a new database. For the steps to restore the backup, see <u>Backing Up and Restoring the VMM Database</u>.

- 4. Update the SCVMM server name in Team Foundation Server.
  - Open a command prompt and run the command: Tfsconfig.exe lab /settings / scvmmservername:my\_new\_scvmmservername /force. You must run TFSConfig from a server that is running the Team Foundation application tier. By default, TFSConfig is located in Drive:\Program Files\Microsoft Team Foundation Server 2010\Tools. You must use the /force switch because the library server and host groups are already bound to a team project collection; changing the SCVMM is not allowed unless you use the /force option. For more information about the TFSConfig Lab Settings command, see TFSConfig Lab /Settings Commands.
  - 2. In the Team Foundation Server Administration Console, click the Lab Management node. Click Reconfigure Lab Management and then click Test near VMM Server name. This confirms that Team Foundation Server can connect to the new SCVMM and has the correct permissions.
    - 1. If the Team Foundation Server service account does not have the appropriate permissions on the specified SCVMM, a dialog box prompts you for a user name and a password.
    - 2. Type the name and the password of a user who is an existing member of the SCVMM Administrator role. The Team Foundation Server service account will be added to the SCVMM Administrator role.
  - 3. In the Team Foundation Server Administration Console, do the following for each team project collection.
    - 1. Click the Lab Management tab and then click Configure Host Groups. Select all host groups, and then click Verify. This confirms that Team Foundation Server and the new SCVMM can connect to the hosts and have the correct permissions.
    - 2. Click , Lab Management, and then click Configure Library shares. Select all library shares and then click Verify. This confirms that Team Foundation Server and the new SCVMM can connect to the library shares and have the correct permissions.

- 5. Confirm the move was successful:
  - 1. In Microsoft Test Manager, click Lab Center and then Environments. Confirm that your existing environments appear in the list and that they are in the correct state.
  - 2. Click Library and then VMs and Templates. Confirm that all the existing virtual machines and templates appear in the list and that they do not display new errors.

## **Move a Test Controller**

If you must move or reinstall a test controller that is associated with environments, possibly for a hardware upgrade or because the current server is becoming too old, you must back up the database and reinstall on the new machine. If you are moving a test controller used for load tests, you back up and restore the load test database additionally. You will also have to go to each agent and reregister it with the new controller.

#### To move or reinstall a test controller

1. Locate the QTControllerConfig.XML file and make a backup copy.

This file is usually located in your Visual Studio installation in the Common7\IDE folder.

- 2. Rebuild the machine with the same name.
- 3. Reinstall the test controller on the new machine.

Do not configure the Test Controller yet.

- 4. Replace the default QTControllerConfig.XML file on the machine with the file that you backed up in step 1.
- 5. Configure the Test Controller and connect it to the Project Collection it was connected before.
- 6. Restart the machine.
- 7. If you change the name of a machine where a test controller is installed and the testing capability is configured for an environment, you must update the name of the test controller in the environments using that test controller. You will use Microsoft Test Manager to modify the environment:
  - 1. Stop the environment.

- 2. Edit the environment, and update the Test Controller setting in the Capabilities page. Select the Test Controller using its new name.
- 3. Restart the environment.

## **See Also**

### Tasks

<u>How to: Change the Library Share for Your Team Project Collections</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
How to: Join the SCVMM Server to another Domain Send Feedback

Occasionally, it might be necessary to change your Team Foundation Server from one active directory domain to another, or from a workgroup to a domain. For more information, see <u>Move Team Foundation Server from One</u> <u>Environment to Another</u>. As part of this task, you should move SCVMM and Lab Management to the new domain.

In order to move an SCVMM server to a domain, or change its domain, you must re-install SCVMM. Therefore, you have to save the virtual machines and templates, and then re-define the environments in the new installation. This topic describes this procedure in more detail.

## **Uninstall SCVMM**

- 1. Remove the hosts and library servers from the SCVMM server.
  - 1. Open the Virtual Machine Manager Administrator Console:

Choose Start, Microsoft System Center, Virtual Machine Manager, Virtual Machine Manager Administrator Console.

- 2. Delete the library servers.
- 3. Delete the host groups.
- 2. Delete all lab artifacts from Microsoft Test Manager Library Center:

In the Lab tab, delete the environments. In the Library tab, delete the virtual machines and the environments.

Because the hosts and library servers are no longer associated with

SCVMM, removing these items from Microsoft Test Manager does not delete the actual virtual machines and VHDs.

- 3. Unconfigure SCVMM from Team Foundation Server.
  - 1. Log on to the Team Foundation Server as tfsadmin.
  - 2. Open Team Foundation Administrator Console.
  - 3. In the Application Tier node, choose Lab Management, then choose Reconfigure Lab Management.
  - 4. Delete the links between the library shares and the team project collection.
- 4. Uninstall SCVMM from the computer on which it was installed.

### Join the computers to domain

- 1. Change the domain of the hosts and library servers.
- 2. Change the domain of the SCVMM server computer.

## **Re-install SCVMM and rebuild artifacts**

- 1. Re-install SCVMM on its computer.
- 2. Add the hosts and library servers back to SCVMM.

For more information, see the SCVMM section in <u>Configuring Lab</u> <u>Management for SCVMM Environments</u>.

3. If any of the virtual machines has to be joined to the new domain, do so.

Network isolated environments have their own domain or workgroup. Therefore, their constituent machines do not have to be altered.

4. Configure the new SCVMM to couple it to your team project collection.

Set the new lab service account in Team Foundation Server.

For more information, see the SCVMM section in <u>Configuring Lab</u> <u>Management for SCVMM Environments</u>.

5. Using Lab Management, re-import the virtual machines and templates into the Library.

For more information, see How to: Import a Virtual Machine or Template from SCVMM.

6. Compose virtual machines into environments in Microsoft Test Manager.

For more information, see How to: Store an SCVMM Environment.

Send feedback on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Managing User Permissions for Lab Management
See Also Send Feedback

You can control the level of access that various members of your team have to Lab Management resources by adding each member to security groups for each team project. By default, Team Foundation creates several groups for each project, and each group has its own set of permissions and rights for that project. If the default groups do not provide the appropriate permissions, you can create custom groups that have a specific combination of permissions. For more information about adding users to the default groups, see Adding and Removing Users To and From Groups. For more information about creating custom groups, see Custom Groups.

# **Default Groups and Permissions**

When you create a team project collection, Team Foundation automatically creates the following default collection-level groups:

- Project Collection Administrators
- Project Collection Valid Users
- Project Collection Service Accounts
- Project Collection Build Service Accounts
- Project Collection Proxy Service Accounts
- Project Collection Test Service Accounts

Of these groups, the Project Collection Administrators and the Project Collection Build Service Accounts are given explicit permissions to access or control Lab Management resources. For more information about all the default collectionlevel groups, see Default Groups in Team Foundation Server.

Similarly, when you create a team project, Team Foundation automatically creates the following default project-level groups:

- Project Administrators
- Contributors
- Readers
- Builders

Of these groups, the Project Administrators, Contributors, and Readers are given explicit permissions to access or control Lab Management resources. For more information about the permissions for all the default collection-level and project-level groups, see <u>Team Foundation Server Permissions</u>.
The following table lists the specific Lab Management permissions that are assigned to users when you add the user to a default group. For many team projects, assigning users to either the Project Administrators group or the Contributors group is sufficient for the individual people to do their jobs.

Name of Lab Management permission	Name of the permission at the command line	Users who have this permission can:	Project Collection Administrators	Projec Collecti Build Servic Accoun
View Lab Resources	Read	View information for the various Lab Management resources, which include collection host groups, project host groups, and environment. To view information about a specific lab resource, you must have the View Lab Resources permission for that resource.	x	X
		Import a		

Import Virtual Machine	Create	virtual machine from a VMM library share. This permission differs from Write because users can create an object in Lab Management but not write anything to the Virtual Machine Manager host group or library share.	X	
Write Environment and Virtual Machines	Write	Users who have this permission for a project host group can create environments. Users who have this permission for a project library share can store environments and templates.	X	X

Edit Environment and Virtual Machines	Edit	Users who have this permission can edit environments and templates. The permission is checked for the object that is being edited.	X	X
Start	Start	Start an environment.	Х	X
Stop	Stop	Stop an environment.	Х	X
Pause Environment	Pause	Pause an environment.	Х	X
Manage Snapshots	ManageSnapshots	Users who have this permission can perform all snapshot management tasks for an environment, which include taking a snapshot, reverting to a	Х	X

#### Delete Environment and Virtual Machine

snapshot, renaming a snapshot, deleting a snapshot, and reading a snapshot.

Delete environments and templates. The permission is checked for the object that is being deleted.

Edit the locations of Lab Management resources, which include collection host groups, collection library shares, project host groups, and project library shares. To change a specific location, you must have the

Manage Lab		Manage Lab
Manage Lab Location	ManageLocation	Manage Lab Locations X permission for that location. This permission for collection- level locations (collection host groups and collection library shares) also lets you create project-level locations (project host group and project library share).
Delete Lab Locations	DeleteLocation	Delete the locations for Lab Management resources, which include collection host groups, collection library shares, project host groups, and project library shares. To delete a

		location, you must have the Delete Lab Location permission for that location.
Manage Child Permissions	ManageChildPermissions	Users who have this permission can change the permissions of all the child Lab Management objects. For example, if a user has Manage Child X Permission for a team project host group, the user can change permissions for all the environments under that team project host group.
		Modify the permissions for a Lab

Manage Permissions	Management object. This permission is X checked for the object whose permissions are being modified.
-----------------------	--

# **Custom Groups and Permissions**

If the permissions granted by the default Team Foundation security groups are too inclusive or exclusive, you can create new security groups that have different combinations of permissions. For example, your team might have some users who function as test leads and other users who function as just testers. The policies in your organization require that only test leads are authorized to create virtual machines and templates. Because adding a tester to the default Contributors group would automatically give the tester permission to create new virtual machines and templates, you might want to create a custom group named "Tester" that has just the permissions indicated in the following table. Similarly, your organization has some users who function as a team project administrator and other users who function as test lab administrators. The policies in your organization require that only team project administrators can manage user permissions. Because adding a lab administrator to the default Project Administrators group would automatically give the test lab administrator the ability to manage permissions, you may want to create a custom group named "Lab Administrators." The following table illustrates the specific permissions that might be given to the "Lab Administrators" group and three other custom groups that differ from the permissions in the default groups. For the steps to create custom groups, see Custom Groups in Team Foundation Server.

Name of Lab Management permission	Custom group: Lab Administrator	Custom group: Test Lead	Custom group: Tester	Custom group: Developer
View Lab Resources	X	Х	Х	Х
Create VM Template using VMM	Х	X		

Virtual Machines				
Write Environment and Template	Х	Х		
Edit Environment and Templates	Х	Х	X	X
Start Environment	Х	Х	X	X
Stop Environment	Х	Х	X	X
Pause Environment	Х	Х	X	X
Manage Environment Snapshots	Х	Х	X	X
Delete Environment and Templates	Х	Х		
Manage Lab	Х			

Location			
Delete Lab Location	X		
Manage Child Permissions			
Manage Permissions			

After you create the custom group, you must add users to those groups. Use the command line utility **TFSLabConfig** to manage the Lab Management permissions for groups or users. For the syntax and other considerations, see <u>Configuring Lab Management with TFSLabConfig</u>.

To view what groups or users have existing permissions on the various Lab Management objects, see the <u>TFSLabConfig Permissions Command</u>.

## See Also

### Concepts

Configuring Lab Management for SCVMM Environments

#### **Other Resources**

Changing Existing Lab Management Configurations

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Planning Your Lab
See Also Send Feedback

As you prepare for widespread use of Lab Management in your test lab, several questions are likely to come to mind:

- How many physical servers do I need?
- What kinds of servers should I buy?
- How much storage capacity do I need?
- Can I use a SAN for storage?
- Can I setup everything on one powerful machine?
- How do I set up an isolated lab?

This topic provides general guidelines for estimating the numbers and types of physical servers, virtual machines (VMs), and controllers you will need for using Visual Studio Lab Management. In general, the number of servers is not as important as the capacity of each server. For example, a server that uses a dual-core or quad-core processor will be able to support more VMs than a server that uses a single-core processor. Similarly, a server that has 32 GB of RAM can host more VMs at the same time than a server that has only 8GB of RAM.

# **Planning for Capacity**

The following sections contain guidelines to help you provide sufficient capacity for your virtual testing lab. The guidelines are expressed as things to do (Do's) and not do (Don'ts) when you acquire or configure hardware or when you install and configure the required software.

### **Planning for Team Foundation Server**

Don't

- Set up everything on a single machine. Only if you will use a single machine just for demonstration or proof of concept purposes should you set up all the components on a single machine.
- Use the NetworkService account as the service account for Team Foundation Server if your instance of Team Foundation Server uses more than one server to run the logical application tier, and you will be using the Lab Management feature of Visual Studio Using the NetworkService account requires more manual work later to maintain the physical host machines for virtual environments. This extra work is necessary because the NetworkService account for each new application-tier machine has to be added to the local Administrator group on each physical host machine. For example, if you run a virtual lab with 20 physical hosts and add or replace an application-tier machine, you would then have to update each of the 20 host machines with the name of the new application-tier machine and assign permissions. Instead of using the NetworkService account, use a standard domain user account and password for the TFSservice account. By doing this, the domain user account is added once at the initial configuration of the physical host and each subsequent application-tier machine uses the same account. For more information about the limitations of the NetworkService account, see <u>NetworkService Account</u>.

## **Planning for System Center Virtual Machine Manager**

Don't

- Install SCVMM on a virtual machine. Installing SCVMM on a virtual machine will make it harder to administer the physical host that virtual machine is running on, and it will slow the performance of the library if you set the library up on the same virtual machine.
- Use clustering with the SCVMM 2008 R2 library servers. Lab Management only supports clustering in SCVMM environments when using SCVMM 2012, not SCVMM 2008 R2.
- Set virtual LAN IDs for network adapters. Lab Management does not support setting and using a virtual LAN ID in System Center Virtual Machine Manager. If you manually set the VLAN ID on a network adapter for a virtual machine and then store the virtual machine in the SCVMM library, the VLAN ID will be cleared when the virtual machine is deployed. For more information about how to use VLAN IDs, see <u>Configuring Virtual Networks in VMM</u>.

#### Do

- Provide the SCVMM machine enough resources. If you expect to have fewer than 50 VMs in your lab, the machine running SCVMM should have at least:
  - A 64-bit processor
  - 4 GB of memory
  - A 300 GB hard disk drive
  - Windows Server 2008 R2 operating system

If you expect to have more than 50 VMs, increase these resources. If you plan to install SCVMM along with some other software on the same machine, give SCVMM server the amount of resources that were described earlier in this topic. However, be sure to determine the amount after you deduct the resource consumption of the other software. For instance, if you want to install SCVMM on the machine that is running Team Foundation Server, add the requirements to those of Team Foundation Server, and then ensure that the machine has enough capacity.

- Provide the server that runs the library at least 200GB of free space on the hard disk drive. In the default installation, make sure that the drive used by the library share has more than 200GB free space.
- Create the default library share on D: and not C:. By default, SCVMM creates the library share on the same machine it is on and creates the library in the C: drive. Changing the default library share to D: makes it easier to upgrade the machine later.
- Use a hard disk drive with sufficient speed for the library. If you plan to use the library lightly, a hard disk with sufficient speed will be sufficient. If you plan to use the library moderately, use a RAID 5 disk configuration with 6 to 12 disks for better throughput. If you plan to use the library heavily, use multiple library servers. You can use direct-attached storage or SAN. When using SAN, create a LUN to be used solely for library machine.
- Run Team Foundation Server under a regular domain user account instead of the network service account. This is required if you put Team Foundation Server and SCVMM on the same machine.
- If SCVMM is installed on a Hyper-V host, store the Hyper-V hosted virtual machines on a different hard disk drive than the SCVMM library. For example, use C: from one disk for the library, and D: from another disk for Hyper-V virtual machines. SCVMM server, in this case, will be running in the primary OS in Hyper-V. This ensures that when the primary OS is loaded, all guest OS (VMs deployed in Hyper-V) will be impacted. To reduce this impact, configure the host reserves for that machine by adding the Hyper-V host reserves (described below) to the SCVMM machine requirements mentioned earlier. Host reserves can be configured using the SCVMM Administrator Console.
- Provide line-of-sight network routing between SCVMM and Team Foundation Server, hosts, and other library servers.
- Update the SCVMM machine with all the latest Windows updates and ensure these updates get applied automatically. If this is not feasible, you should plan to keep track of Windows and SCVMM updates, and apply them manually as they become available.

## ■Planning for the Hyper-V hosts

### Don't

- Install any additional software such as Team Foundation Server on the physical host machine. If you have sufficiently powerful hosts (exceeding the aggregate needs of the hypervisor and virtual machines), then you can have SCVMM or library server co-located on the host, as long as you also account for the resource constraints of those servers. For example, if you want to install SCVMM on a Hyper-V host machine, then add the host requirements, virtual machine requirements, and SCVMM requirements, and then ensure that the machine has enough capacity.
- Use clustering with Hyper-V host servers. Lab Management supports clustering in SCVMM environments.
- Schedule tens of VMs deployments simultaneously. Limit the number of concurrent environment deployments to hosts.
- Use physical hosts that are in different geographic locations that the library servers. If you must use hosts that are in a different geographic location than the SCVMM library servers, the network speed between SCVMM and hosts should be at least 100 Mbps and not subject to high latencies.
- Create multiple network adapters on a virtual machine that connects to a specific network. Lab Management overrides this configuration and creates two adapters. One adapter connects to the lab network and the other adapter handles internal communication between virtual machines.
- Configure the MAC address on a network adapter used in a networkisolated environment. Lab Management clears the MAC address at the time the network-isolated environment is created.

### Do

• Provide the host machines with enough resources and configure them correctly. The number of Hyper-V hosts and the capacity of each host depends on the number of VMs that you host in your lab. If you decide to setup a relatively small lab, install the Hyper-V role on machines with the following configuration:

- Two, dual-core, 64-bit processors that are Hyper-V capable
- 16 GB memory
- 300 GB hard disk space
- Windows Server 2008 R2 operating system
- The latest updates of the Windows operating system.

If you have relatively larger number of virtual machines, and you decide to set up a few, powerful hosts, configure each host as follows:

- two, quad-core, 64-bit processors that are Hyper-V capable
- 64 GB memory
- 1 TB hard disk space
- Windows Server 2008 R2 operating system
- The latest updates of the Windows operating system.
- Reserve enough RAM memory on the host. Out of the host capacity requirements listed above, you must set aside the following resources for the smooth functioning of the hypervisor. For a 16GB host, set aside 20% for the CPU and 2 GB memory. For a 64 GB host, set aside 30% for the CPU and 4 GB memory. These host reserves must be configured in the host properties pane of SCVMM Administrator Console. Only the resources remaining on the host after deducting the host reserves can be used for virtual machines.
- Provide enough storage for virtual machines. You should use a different disk partition for virtual machine storage than the primary partition of Hyper-V server. For example, use D: for virtual machine storage and C: for the primary partition for the hypervisor. After you decide on the virtual machine storage location, configure that location in Hyper-V Manager or using SCVMM Administrator Console. In Hyper-V Manager, change the Virtual Hard Disks folder and the Virtual Machines folder. In the SCVMM Administrator Console, change the Placement Path under the host

properties.

- Provide the hosts with fast hard disk drives and configure the drives correctly. Disk types for hosts: A disk with good speed is necessary. RAID 5 configured disks are highly recommended. The storage for hosts can come from Direct-attached storage or from SAN. However, if you decide to have your host's disk come from a SAN drive for space and reliability needs, you will have to have separate LUNs mapped to each host. Even if the LUNs are managed by same controller, given that Visual Studio Lab Management does not leverage any of SAN functionalities, the underlying BITS copy during a virtual machine deployment will happen all the way from library to host via your LAN network.
- For SCVMM to be installed on a Hyper-V host, it is highly recommended that the hard disk drive used for storing Hyper-V hosted virtual machines is different from the disk used for library. SCVMM server, in this case, will be running in the primary OS in Hyper-V. So, when the primary OS is loaded, all Guest OS (VMs deployed in Hyper-V) will have performance impact. To reduce this impact, configure the host reserves for that machine by adding the Hyper-V machine's host reserves to the SCVMM machine requirements mentioned earlier. Host reserves can be configured using SCVMM administration console.
- For a Hyper-V host to be used as a library server as well, you must have multiple disks in the machine. You should use separate hard disks on the host for the virtual machines and for the library storage.
- Provide the Hyper-V host with line-of-sight networking to Team Foundation Server, SCVMM, and other library servers.
- If the Hyper-V hosts are in different geographic locations, have a local library server for each location as well.
- Update the hosts regularly. Hyper-V hosts should be on a network from where operating system updates can be automatically applied. If this is not feasible, you should plan on keeping track of Windows and SCVMM updates, and apply them manually when they become available.

## **Planning for Controllers and Agents**

Don't

• Install a test controller inside an environment. Only the build, test, and lab agents should be installed on the virtual machines inside an environment.

#### Do

- Use more than one build controller when building and deploying an application for testing. The first controller is used by the build process and is not heavily utilized. The second controller is used to deploy the build to virtual machines and run tests; therefore, it can be heavily used if there are a large number of virtual machines in your lab. The second controller is also used to take snapshots of the environment.
- Use test controllers in the same domain as Team Foundation Server. If esprtfs and a test controller are in a workgroup or untrusted domain, you must create a local user account with same user name and password on both machines, add this user on Team Foundation Server to the "[Project Collection]\Project Collection Test Service Accounts" security group, and then register the test controller with team project collection using this local account.

# **Planning for Topology**

### Do

- Use a gigabit network to connect the server where SCVMM is installed to the library servers and to the Hyper-V hosts.
- Establish a full, two-way trust relationship among the domains where Team Foundation Server, the test controller, the build controller, SCVMM, and the physical host of the virtual machines are running.

There are several topologies you can use when setting up Lab Management for testing your application. The simplest topology for using Lab Management requires only two servers: install all Team Foundation Server components on the same server and install all SCVMM 2008 components on an additional server. Alternatively, you might have complex networking topology requirements that restrict the networks in which Team Foundation Server, SCVMM, Hyper-V hosts, and virtual machines running the application-under-test can be located. In another alternative, you might want to configure network load balancing for your Team Foundation Server. The following list suggests several possible dimensions for your topology and the variations within each dimension.

#### Networking

- DNS
- Firewall
- Threat Management Gateway

#### Domain

- One-way trust
- Two-way trust
- No-trust

Team Foundation Server logical application tier

- Single server
- Multiple servers without network load balancing
- Multiple servers with network load balancing

Team Foundation Server logical data tier

- Single server
- Multiple servers without clustering
- Multiple servers with clustering

#### Tests

- Inside the environment
- Outside the environment

The following four sample topologies are examples of how you can set up combinations of the above dimensions according to your testing needs.

## **■Sample Topology 1**

The Team Foundation Server logical application tier is run on several servers and those servers are controlled by a network load balancer. There is also a separate test network with firewall settings to control the test traffic into and out of the domain network. The following diagram illustrates topology 1.



For instructions to set up this topology, see <u>Setting up various topologies to test</u> with Visual Studio Lab Management - Part 1.

## **■Sample Topology 2**

The Team Foundation Server logical application tier and data tiers are run on several servers, but those servers are not controlled by a load balancer. There is also a separate test network with a SAN-based library and host. The following diagram illustrates topology 2.



For instructions to set up this topology, see <u>Setting up various topologies to test</u> with Visual Studio Lab Management - Part 2.

## **■Sample Topology 3**

The Team Foundation Server logical application tier is run on several servers and those servers are controlled by a network load balancer. There is also a separate test network. The applications being tested make calls to a database outside the virtual environment. The following diagram illustrates topology 3.



For instructions to set up this topology, see <u>Setting up various topologies to test</u> with Visual Studio Lab Management - Part 3.

## **■Sample Topology 4**

The Team Foundation Server logical application tier and data tiers are run on several servers and those servers are controlled by a network load balancer. The test network and environments are in a separate domain. The following diagram illustrates topology 4.



For instructions to set up this topology, see <u>Setting up various topologies to test</u> with Visual Studio Lab Management - Part 4.

# See Also

#### Concepts

Team Foundation Server Architecture Examples of Simple Topology Examples of Moderate Topology Examples of Complex Topology Configuring and Administering Lab Management

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Upgrading from Visual SourceSafe
Send Feedback

Upgrading your code projects, files, version history, labels, and user information from <u>Visual SourceSafe</u> to Team Foundation Server (TFS) or <u>Team Foundation</u> <u>Service</u> version control has many benefits for your team. TFS version control is a modern version control system that is fully integrated with the suite of ALM tools in Visual Studio 2012 and Team Foundation Server.

### To upgrade data from your Visual SourceSafe database

- 1. <u>Understand How VSS Upgrade Tools Convert Your Data</u> Team Foundation and Visual SourceSafe have significant functional differences. As a result, the VSS upgrade tools must modify certain kinds of data as it is upgraded.
- 2. <u>Prepare to Upgrade from Visual SourceSafe</u> Before you start the upgrade process, it's critical that you plan ahead and prepare your data.
- 3. Upgrade your data.
  - <u>Upgrade from Visual SourceSafe Using the Wizard</u> In most cases, you should use the VSS Upgrade Wizard. Advantages include:
    - A fast and easy-to-use step-by-step graphical interface.
    - Support for upgrading into a team project on either an onpremises Team Foundation Server or on <u>Team Foundation</u> <u>Service</u>.
  - <u>Upgrade from Visual SourceSafe Using the VSSUpgrade Command-</u> <u>Prompt Tool</u> You can use the VssUpgrade command-prompt tool to upgrade to an on-premises Team Foundation Server:

- Using a script to upgrade your data.
- While taking advantage of <u>VssUpgrade features</u> that are not available when using the wizard.
- 4. <u>Take Next Steps after Upgrading from Visual SourceSafe</u> After the upgrade tool has finished processing your data, you should verify that the process led to the outcome you expected. You might need to take additional steps to complete the upgrade.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Understand How VSS Upgrade Tools Convert Your Data Send Feedback

The Visual SourceSafe upgrade tools are a convenient method for a one-time, one-way upgrade of your codebase from Visual SourceSafe to Team Foundation Server (TFS). You should use the tools to upgrade your data only when you are ready to stop accessing it in Visual SourceSafe and begin maintaining it in TFS.

TFS and Visual SourceSafe have significant functional differences. As a result, the Visual SourceSafe upgrade tools modify certain kinds of data during the upgrade.

In this topic

- How changesets are created
- How shared and pinned items are upgraded
- How historical data is ported
  - How data about the user name and the time stamp is ported
  - How specific types of events are converted
- How version control bindings are ported

# **How changesets are created**

TFS version control groups changes to multiple files into a single unit when a user checks in a set of changes. This single unit is known as a changeset.

Visual SourceSafe does not have a feature that is equivalent to changesets. However, during the conversion process, each set of changes is grouped into a changeset as long as the following conditions are true:

- The changes do not conflict with each other. For example, no two actions affect the same file or folder.
- The changes occurred within no more than a few minutes of each other.
- The changes were checked in by the same user.
- The changes have the same check-in comment.

# ■How shared and pinned items are upgraded

In Visual SourceSafe, you can share a file across multiple folders. Changes that you make in a shared file are replicated across the folders in which the file is shared. TFS version control does not have an equivalent feature. During the upgrade, shared files in your Visual SourceSafe project are upgraded by creating an additional independent copy of the item on your server for TFS version control.

TFS version control also does not have a feature that is equivalent to the Pin feature in Visual SourceSafe. During the upgrade, pinned items in the Visual SourceSafe project are converted to labeled items on your server for TFS version control. For more information, see the next section.

# **How historical data is ported**

Each event in the history of an item in your Visual SourceSafe database is transferred onto your Team Foundation Server as a changeset. After the upgrade is completed, you can view this data in the History window. For more information, see Get the History of an Item.

Some changes to the data occur during upgrade.

### **How data about the user name and the time stamp is ported**

As each entry in the history of an item in your Visual SourceSafe database is upgraded to a changeset on your Team Foundation Server, the following changes occur:

- The time stamp on the changeset is set to the date and time when the item was upgraded.
- The original time stamp is stored in the Comment field of the changeset.
- The user name is stored in either the User field or the Comment field of the changeset, depending on the result of the user mapping process.

#### ⊠Note

If you want to control how user names are assigned to historical data, you must use VssUpgrade. See Specify how user names are upgraded.

### **How specific types of events are converted**

Events such as edit, rename, and delete are upgraded from your Visual SourceSafe database into changesets on your Team Foundation Server in a straightforward manner. However, the upgrade tools convert some events in ways that you might not expect, as the following table shows.

Visual SourceSafe Event	How it is upgraded into TFS version control
Add file or folder	This changeset is the first event in the history of each upgraded file and folder. Unlike in Visual SourceSafe, no event is logged for the parent of each child item that it contains.
	Sharing is a precondition of branching in Visual SourceSafe, but TFS version control does not support sharing. Therefore, the upgrade of a branched file creates a copy of the file in the destination folder.
Branching	The shared files in your Visual SourceSafe database are upgraded to TFS version control by copying the version of the file that existed when it was shared and putting the copy in the destination folder. Thereafter, each changeset is replicated in both copies of the file until the branch event occurs.

	Implicitly labeled items: In Visual SourceSafe, when you label a folder, the items it contains are implicitly labeled. The upgrade process explicitly labels all the upgraded items in your team project.
Label	Explicitly-labeled files: In Visual SourceSafe, when you explicitly label an individual file, a new version of the file is created. The upgrade process ports the label to the applicable version in TFS version control, but it does not create a new version.

Ι

#### ⊠Note

If your Visual SourceSafe database contains many labels that are applied to many files, the upgrade process may be prolonged. If you want to exclude this data, you must use VssUpgrade and configure it to ignore labels. See <Label migrate="false" />.

The Move Folder event creates a new version of the folder in TFS version

	control.
Move folder	If you are using VssUpgrade, it will not upgrade the complete history of items in moved folders unless both the source and destination folders are upgraded at the same time. See Review and resolve issues found by the analyze feature.
	ĭNote
	If the move folder event is combined with a restore event, the history data may not be upgraded correctly.
Restore	No history data that occurs before a restore event is upgraded.
	TFS version control does not support pinning. The upgrade tools upgrade a pinned file by creating two labels.
	The <b>PINNED_LATEST</b> label is applied to the pinned versions of the pinned files and to the most recent version of the unpinned files. The

PIN and UNPIN	<b>PINNED</b> label is applied to only the pinned versions of the pinned files. After upgrade, the <b>PINNED_LATEST</b> label will retrieve the same files as a Get Latest in Visual SourceSafe. However, one exception is that the <b>PINNED_LATEST</b> label might return different files if events other than checking in occurred after a file was pinned, such as renaming or deleting the file.
Sharing	TFS version control does not support sharing. Shared files in your Visual SourceSafe database are upgraded to TFS version control by copying the version of the file that existed when it was shared and putting the copy in the destination folder. Thereafter, each changeset is replicated in both copies of the file.
	During an upgrade of undelete events of a file or folder, the tools replay the event to create a new version of the file and folder in TFS version

Undelete	control.
File or	
Folder	The upgrade tools create a
	changeset that includes the
	file or folder name, the
	date and time when it was
	undeleted, and the user
	name.
# **How version control bindings are** ported

The upgrade tools port version control bindings for each solution.

# **Next Step**

## Prepare to Upgrade from Visual SourceSafe

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Prepare to Upgrade from Visual SourceSafe
Send Feedback

Before you begin the upgrade process from Visual SourceSafe (VSS) to Team Foundation Server (TFS), plan ahead and prepare your computers and databases to improve the outcome and avoid serious problems.

#### ⊠Note

If your Visual SourceSafe database is a version earlier than Visual SourceSafe 6.0, you must first upgrade it to Visual SourceSafe 2005. Then you can upgrade your data into TFS. This process is explained below.

Prepare to upgrade

- 1. <u>Make sure you have the required permissions</u>
- 2. <u>Schedule the upgrade with your team</u>
- 3. Provide a database for upgrade tool to use
- 4. <u>Prepare your instance of Team Foundation Server</u>
- 5. <u>Copy and prepare your Visual SourceSafe database</u>
- 6. (Optional) truncate the history of items
- 7. <u>Prepare the upgrade computer</u>

# ■Make sure you have the required permissions

To upgrade from VSS to TFS, you must have the following permissions:

- In the VSS database that contains the data that you want to upgrade, you must know the password of the Admin account.
- On the temporary database that the upgrade tool uses, you must be a SQL Administrator or have the CREATE ANY DATABASE permission.
- On your Team Foundation server, you must be a member of the Team Foundation Administrators security group. See

Team Foundation Server Permissions.

• On the upgrade computer-the computer that contains the copy of the VSS database that you are upgrading-you must be a member of the Administrators group.

# **Schedule the upgrade with your team**

Try to schedule the upgrade when your team does not require access to the VSS database that you are upgrading. If you have lots of data, a large team, or if you have worked on the projects for a long time, you should allow time to prepare and upgrade your data.

#### **▲Important**

Inform your team members when the upgrade process will occur, and advise them to check in all files before the process begins.

# Provide a database for the upgrade tool to use

The upgrade tools require a database-either

SQL Server Express or SQL Server-to use for temporary storage. After the upgrade process is complete, the database is never used again. **✓Tip** 

By default, when you install Team Foundation Server Express, SQL Server Express is installed, and you are automatically given the required CREATE ANY DATABASE permission.

# Prepare your instance of Team Foundation Server

Prepare the upgrade computer by following these steps:

- 1. Make sure that the data tier for TFS has enough storage space available. Typically, you will need about two times the data size of the projects in the VSS database that you are upgrading, but the exact amount of storage space depends on the following factors:
  - The size of the VSS database you are upgrading.
  - The number of actions to be upgraded.
- 2. The upgrade tools require that the destination team projects already exist on your Team Foundation server before the upgrade process starts. If you do not yet have the team project collection or the team project into which you want to port your upgraded VSS data, you must create them. See

Create a Team Project Collection and Create a Team Project.

We recommend that you use a new team project that you have not yet begun using. You can upgrade your data into a team project that you are already using. However, if the path to any of your VSS items overlaps with an existing version control path, the upgrade process will fail.

# ■Copy and prepare your Visual SourceSafe database

Copy and prepare your VSS database by following these steps:

- 1. Check in files. Ideally, all files in your VSS database should be checked in. If this is not possible, as many files as possible should be checked in before you upgrade.
- 2. Remove access to the Visual SourceSafe projects. You should be the only person who has access to the Visual SourceSafe projects that you are upgrading.
- 3. Copy the database. Follow the instructions on this page of the Microsoft Web site:

How To Back Up a Visual SourceSafe Database.

• Upgrade the copy of your database. If your Visual SourceSafe database is a version earlier than Visual SourceSafe 6.0, upgrade it to Visual SourceSafe 2005 by using the Visual SourceSafe DDUPD Utility.

• Scan for and fix data integrity issues in the copy of your database.

## **▲Important**

You must use the Visual SourceSafe ANALYZE utility to locate and fix data integrity issues in the database. For more information about how to use this tool, see the following pages on the Microsoft Web site: <u>ANALYZE Utility</u> and <u>How to Detect and Fix Database Corruption Errors in Visual SourceSafe</u>.

# Optional) Truncate the history of items

If you do not need all the history data, you can save time when you upgrade from Visual SourceSafe and preserve only the history after a specific date. This is called truncating the history. To do this, use the Archive feature in VSS.

#### ĭTip

If you want to exclude all historical data and you plan to use the VSS Upgrade Wizard, you can skip this section.

#### **▲**Caution

Archiving permanently removes the version history from the VSS database. Therefore, make sure that you perform this procedure on a copy of the VSS database instead of the database that is in service.

You can specify the time stamp before which you want to truncate the history by using any of the following values:

- Label
- Version of a folder
- Date

For more information about how to archive in Visual SourceSafe, see Visual SourceSafe Archive Databases.

#### ⊠Note

The Visual SourceSafe Archive feature has a limitation of 2 gigabytes (GB) on the size of the archive file. If an error occurs while you are archiving, try to archive smaller projects separately.

## **Prepare the upgrade computer**

Log on and prepare the computer on which you will perform the upgrade:

- 1. Make sure the computer is running one of the following operating systems:
  - Windows 8
  - Windows Server 2012
  - Windows 7
  - Windows Server 2008 R2
- 2. Install the

Visual SourceSafe upgrade tools.

• Make sure that the computer has sufficient free disk space to complete the upgrade process. To estimate how much disk space is required, total the following items:

- 5 GB for the upgrade tools to create temporary files and to generate log files.
- Two times the size of the projects in the Visual SourceSafe database that you will upgrade.

• Make sure that you have followed the steps in <u>Copy and Prepare Your Visual</u> <u>SourceSafe Database</u> earlier in this topic.

• Copy the VSS database to a folder on the upgrade computer.

#### ⊠Note

If you use file sharing to enable the upgrade computer to access the data in the VSS database instead of copying the database, you must provide Read and Modify access to the account that you use to log on to the upgrade

computer. This approach is not recommended because it may prolong the upgrade process.

#### **A**Caution

Regardless of how you set up your upgrade computer to access your VSS database, make sure that you run the upgrade process on a copy of the database and not the database that is in service. This approach helps protect your data.

## **Next step**

Choose one of the following processes:

- <u>Upgrade from Visual SourceSafe Using the Wizard</u> In most cases, you should use the VSS Upgrade Wizard. Advantages include:
  - A fast and easy-to-use step-by-step graphical interface.
  - Support for upgrading into a team project on either an on-premises Team Foundation Server or on <u>Team Foundation Service</u>.
- <u>Upgrade from Visual SourceSafe Using the VSSUpgrade Command-</u> <u>Prompt Tool</u> You can use the VssUpgrade command-prompt tool to upgrade to an on-premises Team Foundation server:
  - Using a script to upgrade your data.
  - While taking advantage of <u>VssUpgrade features</u> that are not available when using the wizard.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Upgrade from Visual SourceSafe Using the Wizard Send Feedback

After you understand how the upgrade tools work and have completed preparations, you can proceed to upgrade your Visual SourceSafe data. In most cases, you should use the VSS Upgrade Wizard. Advantages include:

- A brief and simple step-by-step procedure to initiate the upgrade.
- Support for upgrading into a team project on either an on-premises Team Foundation Server or on <u>Team Foundation Service</u>.

# **Upgrade your Visual SourceSafe** data using the wizard

#### **▲Important**

You must run Analyze on your Visual SourceSafe repository before you use the wizard. See Copy and prepare your Visual SourceSafe database.

- 1. From Windows Start, run VSS Upgrade Wizard.
- 2. On the Visual SourceSafe Repository page...

Provide and check the Visua Server.	SourceSafe information required for upgrade to Team	Foundation
Type in a Visual SourceSafe reposito or more projects from the list for up	ry folder path and administrator password, click List Available Projects, grade.	and select one
Visual SourceSafe Repository:	C:\VSS	Browse
	Use the <b>Browse</b> button to find a Visual SourceSafe repository folder that has to include the srcsafe.ini file.	
Visual SourceSafe Admin password:		
	The field is optional. Leave it empty if the Visual SourceSafe administrator account does not require a password.	

Specify the repository, and if necessary, the Admin password.

3. To display the projects in your Visual SourceSafe repository, choose the List Available Projects link.

List Available Projects		

Select the projects that contain the data you want to upgrade.

4. Select the check box at the bottom of the page to confirm you have run Analyze. See Copy and prepare your Visual SourceSafe database.

Choose Next.

5. On the Team Project page...



Choose Browse and then use the Select a Team project for Migration dialog box to specify the team project into which you want to port the upgraded data.

We recommend that you use a new team project that you have not yet begun using. You can also upgrade your data into a team project that you are already using. However, if the path to any of your Visual Source Safe items overlaps with an existing version control path, the upgrade process will fail.

Choose Next.

6. On the options page...



Select whether you want to upgrade the Full history or Tip to omit historical data.

#### ĭTip

If you want to upgrade a specific subset of your historical data, you can truncate it before you upgrade. Choose Cancel, and then see Truncate the history of items.

7. On the options page...



Specify the name of the SQL Server instance you want the wizard to use for temporary storage. See Provide a database for upgrade tool to use.

Choose Next.

- 8. Review the configuration settings and then choose Next.
- 9. After the readiness checks all pass, choose Upgrade.
- 10. The wizard analyzes your Visual SourceSafe data, and then upgrades and ports it to your Team Foundation Server. After the process is complete, choose Next.
- 11. View and verify the results, and resolve issues as necessary.

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Upgrade from Visual SourceSafe Using the VSSUpgrade Command-Prompt Tool

Send Feedback

In most cases, you should use the VSS Upgrade Wizard instead of the VssUpgrade command-prompt tool to upgrade your Visual SourceSafe data. However, there are <u>a few situations</u> when you might want to use the VssUpgrade command-prompt tool to upgrade to an on-premises Team Foundation server (TFS).

## To upgrade your Visual SourceSafe data using VSSUpgrade

- 1. Understand how the tool converts your data TFS and Visual SourceSafe have significant functional differences. As a result, the Visual SourceSafe upgrade tools modify certain kinds of data during the upgrade.
- 2. Prepare to upgrade Before you start the upgrade process, it's critical that you plan ahead and prepare your data.

### **▲Important**

You must run the Visual SourceSafe ANALYZE utility on your Visual SourceSafe repository before you proceed. See Copy and prepare your Visual SourceSafe database.

- 3. <u>Learn about unique VssUpgrade features</u>. If you don't need any of them, then you should use the VSS Upgrade Wizard instead.
- 4. <u>Analyze your data</u> Before you upgrade your data from Visual SourceSafe to Team Foundation version control, you must use the Analyze command to determine whether any issues in your data will affect the outcome of the

upgrade. This process also generates a user mapping file that is required to upgrade your data.

- 5. <u>Upgrade your data</u> To upgrade your data, you must specify how user names are upgraded, create a Migrate settings file, and then run the Migrate command.
- 6. View and verify the results, and resolve issues as necessary

# **VssUpgrade features**

Use VssUpgrade if you want to upgrade and port your data to an on-premises team project and:

• Perform the upgrade as part of an automated approach, such as a script.

•

<u>Migrate multiple Visual SourceSafe projects into a single team project</u>, using a folder structure that is specified by you.

• <u>Specify how user data is converted</u>, for example, from Kim in Visual SourceSafe to EUROPE\KimT in TFS.

• <u>Ignore labels</u> in the Visual SourceSafe data.

• <u>Resume the process after it has been interrupted</u> while upgrading a large amount of data-for example, because of a network error.

If you don't need any of these features, or if you need upgrade and port your data into a team project on <u>Team Foundation Service</u>, then you should use the VSS Upgrade Wizard instead of VssUpgrade.

# Analyze your data

Before you upgrade your data from Visual SourceSafe to TFS version control, you must first use the Analyze command to determine whether any issues in your data will affect the outcome of the upgrade. This command also generates a user mapping file that the Migrate command uses to upgrade your data.

## **Create an Analyze settings file**

Before you run the Analyze command, you must create an Analyze settings file. In this file, you specify the path of the Visual SourceSafe database that you will upgrade and the folders that you want to upgrade.

The following XML is an example of an Analyze settings file.

#### Copy Code

```
<?xml version="1.0" encoding="utf-8"?>
<SourceControlConverter>
<ConverterSpecificSetting>
     <Source name="VSS">
          <VSSDatabase name="c:\ourvss"></VSSDatabase>
          <UserMap name="c:\ourvss\migrate\Usermap.xml"></UserMap>
     </Source>
     <ProjectMap>
          <Project Source="$/Core"></Project>
          <Project Source="$/ProjectA"></Project>
          <Project Source="$/ProjectB"></Project>
     </ProjectMap>
</ConverterSpecificSetting>
<Settings>
     <Output file="c:\ourvss\migrate\logs\ContosoVSSAnalyze.xml"></(
</Settings>
</SourceControlConverter>
```

You can copy the previous example, paste it into your own Analyze settings file, and then modify it. The following information can help you adapt the example to meet your needs.

#### **■<?xml encoding> attribute**

The <?xml encoding> attribute must match the encoding that is used in your Analyze settings file. For example, if the file is saved as Unicode, the <?xml encoding> tag is as follows:

```
<?xml version="1.0" encoding="unicode">
```

#### **■**<VSSDatabase name> attribute

In the <VSSDatabase name> attribute, specify the path of the folder that contains the **srcsafe.ini** file for the copy of the Visual SourceSafe database that you are upgrading. For example:

#### Copy Code

```
<Source name="VSS">
...
<VSSDatabase name="c:\ourvss"></VSSDatabase>
...
</Source>
```

The path must not contain the string **srcsafe.ini**. For example, the following <VSSDatabase name> attribute is incorrect and will cause the **VssUpgrade** command to fail:

#### Copy Code

```
<Source name="VSS">
...
<VSSDatabase name="c:\ourvss\srcsafe.ini"></VSSDatabase>
...
</Source>
```

#### **=**<UserMap name> attribute

The Analyze command gathers and compiles data about your Visual SourceSafe users and stores it in an XML file. Optionally, you can specify the path and name of the file where you want this data stored in the <UserMap name> attribute. If

you do not specify this attribute, the Analyze command creates a file that is called **UserMap.xml** and puts it in the current directory.

#### **■**<**ProjectMap**> section

In the <ProjectMap> section, specify the path of each Visual SourceSafe project that you want to upgrade in the Source attribute of a <Project> item.

To upgrade all the data in your Visual SourceSafe database, make the <ProjectMap> section match the following example:

#### Copy Code

```
<ProjectMap>
<Project Source="$/"></Project>
</ProjectMap>
```

Instead of upgrading your complete Visual SourceSafe database at the same time, you can upgrade selected projects at different times.

#### ĭTip

Use this option to avoid blocking your team during upgrade if you have lots of data to upgrade.

The paths in the Source attributes must not overlap. For example, the following <ProjectMap> section is not valid:

#### Copy Code

```
<ProjectMap>
<Project Source="$/ProjectA"></Project>
<Project Source="$/ProjectA/Controller"></Project>
</ProjectMap>
```

**■**<Output file> attribute

In the <Settings> section, in the <Output file> attribute, you can specify the path and name of the file where you want the analysis report to be written. If you do not want to specify this option, you can omit the <Output> tag. In this case, the converter writes the report to a file that is named **VSSAnalysisReport.xml** and puts it in the current directory.

### **=<SQL> element**

You must specify a database for VssUpgrade to use to store temporary data by adding a <SQL> element to the <Source> section of your Analyze settings file. This element uses the following syntax: <SQL Server="SQL\_Server\_name"> </SQL>.

For example, if you are running VssUpgrade on a computer that is named **FABRIKAM-2**, to use a local SQL Server Express instance on the same computer:

#### Copy Code

```
<Source name="VSS">
...
<SQL Server="fabrikam-2\sqlexpress"></SQL>
...
</Source>
```

To direct the converter to use SQL Server, for example a server that is named **FabrikamSQLServer** 

#### Copy Code

```
<Source name="VSS">
...
<SQL Server="fabrikamsqlserver"></SQL>
...
</Source>
```

### **Run the Analyze command**

1. From Windows Start, run Command Prompt as administrator.

2. In the Administrator: Command Prompt window, enter:

VssUpgrade Analyze settings.xml

Replace settings.xml with the path and name of the Analyze settings file that you created.

3. When you are prompted, enter the administrator password for your Visual SourceSafe database. If your database has no password, press Enter.

VssUpgrade displays the ongoing status as the Analyze command proceeds. When the process is completed, the system summarizes the results. For example:

### Copy Code

Analyze complete. Analyzed 859 files and 941 folders. Warnings 0 and Errors 0 Pre-migration report file: C:\VSS\migrate\VSSAnalysisReport.xml

If there were any warnings or errors, you can find details about them in **ConverterErrors.txt** and **VSSUpgrade.log**.

The Analyze command generates:

- A report that you can use to get details about possible problems and changes that the Migrate command may cause.
- A user mapping file (**Usermap.xml**), which you can use to specify how user data will be converted during the upgrade process.

These files are explained in more detail below.

## **Review and resolve issues found by the Analyze command**

The analysis report provides information about issues in your Visual SourceSafe database that may cause problems during the upgrade process. Try to resolve as many of these issues as possible to minimize problems with the upgrade process, as described in the next section.

#### **■Some Files are Checked Out**

The report lists files that are currently checked out. The upgrade process does not preserve checkout information. Try to ensure that as many files as possible are checked in before upgrading.

#### **Some Items have Data Integrity Issues**

The report lists items whose data integrity has been compromised. The Visual SourceSafe ANALYZE utility may be able to fix these kinds of issues. Items that cannot be repaired will not be upgraded. For more information, see the following pages on the Microsoft Web site:

<u>ANALYZE Utility</u> and <u>How to Detect and Fix Database Corruption Errors in</u> <u>Visual SourceSafe</u>.

# ■Some Folders in Mapped Projects Contain History that is Not Included in the <ProjectMap> section

If a folder is moved from one project to another in a Visual SourceSafe database, the history of that folder is contained in both the original and current projects. To upgrade such a folder with all of its history, you must upgrade both the original and current projects.

For example, you are upgrading the Visual SourceSafe project Project2. This project contains the folder \$/Project2/FeatureA, which was moved from Project1 at some point in its history.

If your <projectmap> section contains</projectmap>	For example	Then
	Copy Code	The folder is
Both projects.	<projectmap> <project source="\$/Project1"></project></projectmap>	upgraded

	<project sour<br=""></project>	ce="\$/Project2">	with its full history.
The project that originally contained the folder but not the project that currently contains it.	<projectmap> <project sour<br=""></project></projectmap>	<pre>Copy Code ce="\$/Project1"&gt;</pre>	The folder is not upgraded.
The project that currently contains the folder but not the project that originally contained it.	<projectmap> <project source<br=""></project></projectmap>	<pre>Copy Code ce="\$/Project2"&gt;</pre>	The folder is upgraded with its history starting from when it was moved to the current project. The history that occurred before the folder was moved to the current project is not

For more information about the <ProjectMap> section of the settings file, see <<u>ProjectMap> Section</u> earlier in this topic.

#### **Some label names are not supported by Team Foundation version control**

The report lists label names that will change when they are upgraded because they contain characters that TFS version control does not support. See <u>Naming</u> <u>Restrictions in Team Foundation</u>.

# **Upgrade your data**

After you have run the Analyze command, you are almost ready to upgrade your data. Before you run the Migrate command, you must create a settings file. Optionally, you can specify how user names are upgraded.

## **Specify How User Names are Upgraded**

You can control how user information is upgraded from Visual Source Safe to TFS version control. Specifically, you can specify what user name the Migrate command should associate with each changeset in the history of each item in TFS version control. You do this by editing the user mapping file that was created when you ran the Analyze command, as explained earlier in this topic.

The user mapping file is optional. If you omit the <UserMap name> attribute from your Analyze settings file, each changeset is built in the following manner:

- The User field is set to the name of the account under which VssUpgrade is running.
- The name of the user who performed the action in your Visual SourceSafe database is stored in the Comment field.

#### **■Example of a user mapping file**

When you run the Analyze command, it compiles data about your Visual SourceSafe users and stores it in an XML file. This file lists every Visual SourceSafe user who has ever performed a version control operation in the Visual SourceSafe projects that you are upgrading.

The following example shows a user mapping file that was created by the Analyze command.

#### Copy Code

```
<?xml version="1.0" encoding="utf-8"?>
<UserMappings>
```

```
<UserMap From="Admin" To=""></UserMap>
<UserMap From="Guest" To=""></UserMap>
<UserMap From="Kim" To=""></UserMap>
<UserMap From="Satomi" To=""></UserMap>
<UserMap From="Mark" To=""></UserMap>
</UserMappings>
```

You can specify the To attribute of none, some, or all of the UserMap items in the user mapping file. For example, you could modify the previous example in the following manner:

#### Copy Code

```
<?xml version="1.0" encoding="utf-8"?>
<UserMappings>
<UserMap From="Admin" To="NORTHAMERICA\KenM"></UserMap>
<UserMap From="Guest" To="Test1"></UserMap>
<UserMap From="Kim" To="EUROPE\KimT"></UserMap>
<UserMap From="Kim" To="EUROPE\KimT"></UserMap>
<UserMap From="Satomi" To="ASIA\SatomiH"></UserMap>
<UserMap From="Mark" To=""></UserMap>
</UserMappings>
```

Notice that in the previous example, Guest is mapped to Test1, and no domain is specified. In these cases, VssUpgrade presumes that the account belongs to the default domain.

If you do not specify a <UserMap To> attribute, each changeset is built in the following manner:

- The User field is set to the name of the account under which VssUpgrade ran.
- The name of the user who performed the action in your Visual SourceSafe database is stored in the Comment field.
- If you specify a <UserMap To> attribute and the value is a valid user on your Team Foundation Server, the User field is set to the name of that account. If the value is not a valid user on your Team Foundation Server, VssUpgrade will display an error and end the upgrade process.

## **Create a Migrate settings file**

You use the Migrate settings file to specify what Visual SourceSafe data you want to upgrade and to control several aspects of how you want to upgrade it. The easiest way to create this file is to copy the file that you created in

<u>Create an analysis settings file</u> earlier in this topic. You then add more data to the file to make it usable by the Migrate command.

The following example shows a Migrate settings file.

```
<?xml version="1.0" encoding="utf-8"?>
<SourceControlConverter>
<ConverterSpecificSetting>
     <Source name="VSS">
          <VSSDatabase name="c:\ourvss"></VSSDatabase>
          <UserMap name="c:\ourvss\migrate\Usermap.xml"></UserMap>
     </Source>
     <ProjectMap>
          <Project Source="$/Core" Destination="$/CoreTeamProject"><
          <Project Source="$/ProjectA" Destination="$/ClientTeamPro;
          <Project Source="$/ProjectB" Destination="$/ClientTeamPro-
     </ProjectMap>
</ConverterSpecificSetting>
<Settings>
     <TeamFoundationServer name="My_Server" port="8080" protocol="ht
     <Output file="c:\ourvss\migrate\logs\ContosoVSSMigrate.xml"></(
</Settings>
</SourceControlConverter>
```

The following information can help you modify the Migrate settings file to specify how the Migrate command will upgrade your data.

#### **=<ProjectMap>**

For each <Project> element in the <ProjectMap> section of your Migrate settings file, provide a Destination attribute to specify the path of the location on your Team Foundation Server where you want to upgrade the contents of the project in your Visual SourceSafe database (specified in the Source attribute).

For example, you want to upgrade the contents of ProjectA in your Visual SourceSafe database into ProjectA at the root of a team project that is called

Client.

#### Copy Code

```
<ProjectMap>
<Project Source="$/ProjectA" Destination="$/ClientTeamProject/Prc
</ProjectMap>
```

For the value in the Destination attribute to be valid, the following conditions must be true:

- The team project in the Destination attribute (in the previous example, the team project is ClientTeamProject) must already be located in the team project collection before you start the upgrade process.
- The path in the Destination attribute of a <Project> element must not overlap the path in the Destination attribute of any other <Project> elements. For example, the following <ProjectMap> section is not valid:

#### Copy Code

```
<ProjectMap>
<Project Source="$/ProjectA" Destination="$/ClientTeamProjec
<Project Source="$/ProjectB" Destination="$/ClientTeamProjec
</ProjectMap>
```

#### ⊠Note

If the folder in the Destination attribute contains any items, and if the path to any of your Visual Source Safe items overlaps with the version control path to any of these items, the upgrade process will fail.

#### **∃**<TeamFoundationServer> tag

In the <Settings> section, add a <TeamFoundationServer> tag, and specify the name, port, protocol, and path to the team project collection on your Team Foundation Server by using the following format:

<TeamFoundationServer name="ServerName" port="PortNumber" protocol='

#### **■**<Label migrate="false" /> tag

If your Visual SourceSafe database contains many labels that are applied to many files, the upgrade process may be prolonged. If your team does not need this data, you can configure VssUpgrade to ignore labels by adding the <Label migrate="false" /> tag to the <Settings> section.

#### **=<Output file> attribute**

In the <Settings> section, in the <Output file> attribute, you can specify the path and file where you want the upgrade report written. If you do not include the attribute, the converter writes the report to a file that is named **VSSMigrationReport.xml** and puts it in the current directory.

## **Run the Migrate command**

- 1. From Windows Start, run Command Prompt as administrator.
- 2. In the Administrator: Command Prompt window, enter:

#### VssUpgrade Migrate settings.xml

Replace settings.xml with the path and name of the Migrate settings file that you created.

The Migrate command displays each project that you are upgrading from your Visual SourceSafe database and each folder into which the data will be upgraded on your server for Team Foundation version control.

- 3. When you are prompted, enter the administrator password for your Visual SourceSafe database. If your database has no password, press Enter.
- 4. The system summarizes how it will run the upgrade process and asks you to confirm that you want to proceed. For example:

#### Copy Code

This will start migration with following inputs:

SourceSafe Folders -> Team Foundation Server Folders

\$/ -> \$/Sequence2

Full history migration of VSS Database: c:\vss

Team Foundation Server: http://fabrikam-4:8080/tfs/DefaultColle Migration Settings File: migrate\_settings.xml Migration Report: C:\VSS\migrate\VSSMigrationReport.xml

Depending on the VSS database size, migration may take few hour: Please verify all inputs are correct and confirm. Start migration (Y/N)?

Press **Y** to proceed.

VssUpgrade displays the ongoing status as the Migrate command proceeds. When the process is completed, the system summarizes the results. For example:

#### **₽C**0

Verification of pinned versions: 0 file(s) added, 0 file(s) edited & Verification of latest tip versions: No change was required (No fixe

Started: Creating label "Pinned\_Latest". Finished: Creating label "Pinned\_Latest".

Migration complete. Migrated 6397 Actions Warnings 7 and Errors 0 Post migration report file: C:\VSS\migrate\VSSMigrationReport.xml

If there were any warnings or errors, you can find details about them in **ConverterErrors.txt** and **VSSUpgrade.log**.

The Migrate command generates a report that you can use to get details about possible problems and changes that the Migrate command may have caused. See

Take Next Steps after Upgrading from Visual SourceSafe.

## **Resume the process by using incremental upgrade**

If the upgrade process is interrupted for some reason, you can resume the process as an incremental upgrade from the point at which the process ended. An incremental upgrade can be useful if the upgrade process failed because of an error or network problems. During incremental upgrade, the converter will upgrade only the data that was not upgraded in previous sessions.

To start an incremental upgrade, follow the steps in

<u>Run the Migrate command</u>. When the Migrate command asks whether you want to perform an incremental migration, press **Y**.

Limitations of an incremental upgrade

An incremental upgrade will not succeed unless you comply with the following restrictions:

- In your Visual SourceSafe database, you must not have performed destroy, purge, archive, or restore activities.
- You must not change the <ProjectMap> section of your Migrate settings file.
- On your Team Foundation Server, you must not modify any folders (or any content in the folders) that are specified in the <ProjectMap> section of your Migrate settings file.

## **Next step**

View and verify the results, and resolve issues as necessary

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Take Next Steps after Upgrading from Visual SourceSafe
Send Feedback

After the upgrade is finished, you should make sure the results meet your expectations and that the data is ready for your team to use:

- <u>Check the outcomes from the upgrade process</u>
- <u>Resolve problems</u>
  - <u>Resolve failure caused by storage limit for SQL Server Express</u>
  - <u>Convert files in the MS-DOS-Compatible short name (8.3) format</u> (TF227014)
# ■Check the outcomes from the upgrade process

Review the report and, if necessary, the logs that were generated by the upgrade tool. Check the data on your Team Foundation server to make sure that the data from your Visual SourceSafe database was upgraded in the way that you expected.

### **Wiew the Migration Report**

TFS and Visual SourceSafe have significant functional differences. As a result, the Visual SourceSafe upgrade tools modify certain kinds of data during the upgrade. And in some cases, errors or problems could occur. The migration report can advise you about these issues.

To open the migration report:

- If you used the VSS Upgrade Wizard: On the Complete page, choose the Upgrade Report link.
- If you used the VssUpgrade tool: Open the **VSSAnalysisReport.xml** report in the current directory, or whatever file name and location you specified using the <Output file> attribute.

### **Migration Report**

Status	Migration completed. { 0 Errors   7 Warnings } Click here for post migration steps
Type of Migration	Full
Source Control	Visual SourceSafe
VSS Location	C:\VSS\srcsafe.ini
VSS Folder	Team Foundation Folder
\$/Coverter	\$/Sequence/Coverter
Warnings	
Warnings Label Name Change TF227033: Label name supported by Team Fo	Warnings s were changed as a result of migration because they contain characters not pundation version control, or they contain more than 64 characters. Team
Warnings Label Name Change TF227033: Label name supported by Team Fo VSS Label	Warnings s were changed as a result of migration because they contain characters not bundation version control, or they contain more than 64 characters. Team Team Foundation Label
Warnings Label Name Change TF227033: Label name supported by Team Fo VSS Label Labeling Version No. 3	Warnings         es were changed as a result of migration because they contain characters not pundation version control, or they contain more than 64 characters. Team         Team Foundation Label         0       Labeling Version No. # 0
Warnings Label Name Change TF227033: Label name supported by Team Fo VSS Label Labeling Version No. 3 Labeling Version No. 3	Warnings         es were changed as a result of migration because they contain characters not bundation version control, or they contain more than 64 characters. Team         Team Foundation Label         0       Labeling Version No. # 0         1       Labeling Version No. # 1
Warnings Label Name Change TF227033: Label name supported by Team Fo VSS Label Labeling Version No. 3 Labeling Version No. 3	Warnings         Iss were changed as a result of migration because they contain characters not bundation version control, or they contain more than 64 characters. Team         Team Foundation Label         0       Labeling Version No. # 0         1       Labeling Version No. # 1         2       Labeling Version No. # 2

### **View logs**

If warnings or errors were reported, you might want to view the logs. You might also want to view the logs to get details on how your data was modified during the upgrade process. To view the logs:

- If you used the VSS Upgrade Wizard: On the Complete page, choose the Complete Log link.
- If you used the VssUpgrade tool: Open **ConverterErrors.txt** and **VSSUpgrade.log**.

### **Examine the data in your team project**

Use source control explorer to examine the data on your Team Foundation server to make sure that the data from your Visual SourceSafe database was upgraded in the way that you expected.

# **Resolve problems**

After you have examined the outcome of the upgrade process, you might have to troubleshoot problems.

### **Resolve failure caused by storage limit for SQL Server Express**

The upgrade tools use a database to store temporary metadata. This metadata typically requires a small percentage of the total size of the data that you upgrade. In the unlikely event that your upgrade fails because the 4-GB limit of SQL Server Express is reached, you can use SQL Server instead of SQL Server Express. See

<u>Upgrade from Visual SourceSafe Using the Wizard</u> or VssUpgrade <SQL> Element.

### ■Convert files in the MS-DOS-Compatible short name (8.3) format (TF227014)

TFS does not allow file names that are in the MS-DOS-compatible short name (8.3) format (for example, abcdef~1.txt). When you analyze or attempt to upgrade files that have such a name, a TF227014 error appears.

To work around this issue, you can temporarily apply a setting to your Team Foundation Server that will cause it to allow files that have such names. To do this, you must set **Allow8Dot3Paths** to **True** in the configuration database for TFS.

#### **▲Important**

To avoid issues with client machines that support MS-DOS-compatible short names, after you complete the upgrade process, it is strongly recommended that you set **Allow8Dot3Paths** to **False** as described in the following procedure.

To perform the following procedure, Windows PowerShell must be enabled on

the application-tier server for TFS. See

Scripting with Windows PowerShell.

**Required Permissions** 

You must be a member of the Administrators group on the application-tier server for TFS. For more information, see <u>Team Foundation Server Permissions</u>.

# To upgrade a Visual SourceSafe database that contains files that are named in the MS-DOS-compatible short name format

- 1. Log on to the application-tier server for Team Foundation.
- 2. Create a Windows PowerShell script that is called **Allow8Dot3Paths**.
  - 1. Copy the text in <u>Allow8Dot3Paths PowerShell Script</u> later in this topic, and paste the text into the script.
  - 2. Change ServerPath to match the path in the URL that you use to connect to Team Foundation Server. By default, the server path is "tfs".
  - 3. Change CollectionName to match the name of the team project collection into which you are upgrading your data (for example, DefaultCollection).

The end result, for example, would be the following line in the script:

### Copy Code

\$collectionBaseUrl = "http://localhost:8080/tfs/DefaultColl

- 3. Run the **Allow8Dot3Paths** script.
- 4. Recycle the application pool for your Team Foundation Server.
  - 1. From Windows Start run Computer Management.

- 2. In the navigation pane, expand Services and Applications.
- 3. Choose Internet Information Services (IIS) Manager, expand the local computer, and double-click Application Pools.
- 4. Open the shortcut menu of the application pool, and then choose Recycle.
- 5. Run the Migrate command.
- 6. Modify the **Allow8Dot3Paths** Windows PowerShell script that you created earlier, replacing "true" with "false".
- 7. Run the modified **Allow8Dot3Paths** script.
- 8. Recycle the application pool for your Team Foundation Server (as explained above).
- 9. In Visual Studio, if you are not already connected, then connect to the team project into which you upgraded the data.
- 10. In Source Control Explorer, rename any files that have names in the MS-DOS-compatible short name (8.3) format.

#### ■Allow8Dot3Paths PowerShell Script

#### **₽Co**

# Load client OM assembly. [Reflection.Assembly]::Load("Microsoft.TeamFoundation.Client, Versic \$collectionBaseUrl = "http://localhost:8080/ServerPath/CollectionNam \$tfs = [Microsoft.TeamFoundation.Client.TeamFoundationServerFactory] \$collectionHive = \$tfs.GetService([Microsoft.TeamFoundation.Framewor # Set some version control settings in the collection hive. \$collectionHive.SetValue("/Service/VersionControl/Settings/Allow8Dot # Display all version control settings as a table. \$collectionHive.ReadEntries("/Service/VersionControl/Settings/...") Send <u>feedback</u> on this topic to Microsoft.

Visual Basic 
C#
Visual C++
F#
HLSL
JScript
Technical Reference for Team Foundation
See Also Send Feedback

You can perform many of the management and configuration tasks for Visual Studio Team Foundation Server at a command prompt. You can also use command-line tools to perform tasks for version control and build. This technical reference provides you with a list of all of the command-line tools that are available in Team Foundation Server.

You can also customize the work item types and process guidance that is available to your users in team projects. This technical reference provides details of the schemas for work item types and process guidance. Review the information that is provided about the schemas before you make any customizations.

## **Common Tasks**

Title

#### Description

Manage or script the configuration of your server: You can manage many aspects of your deployment of Team Foundation Server from the command line. You can use the commandline tools for Team Foundation Server to script changes or updates to the deployment.

Work with version control. You

- <u>Managing Server</u> <u>Configuration with</u> <u>TFSConfig</u>
- <u>Changing Groups</u> <u>and Permissions</u> <u>with TFSSecurity</u>
- <u>Configuring Lab</u> <u>Management with</u> <u>TFSLabConfig</u>
- <u>Specify the Bug</u> <u>Type to File By</u> <u>Using Microsoft</u> <u>Test Manager</u>
- <u>Delete a Team</u> <u>Project</u> [TFSDeleteProject]
- <u>Upload or</u> <u>Download the</u> <u>Microsoft Project</u> <u>Mapping File</u> [TFSFieldMapping]

can manage version control and perform tasks from a command prompt.	• <u>Team Foundation</u> <u>Version Control</u> <u>Command</u> <u>Reference</u>
Manage your builds. From a command prompt, you can perform tasks such as starting and stopping builds, in addition to deleting or destroying completed builds.	• <u>Team Foundation</u> <u>Build Commands</u>
Import, export, and manage objects for tracking work items. By using the <b>witadmin</b> command- line tool, you can create, delete, import, and	• <u>witAdmin:</u>

export categories, global lists, types of links, types of work items, and work item fields. You can manage these objects for a team project collection or a specific team project.	Customize and Manage Objects for Tracking Work Items
Look up the definition for an element in a work item type. You can customize types of work items, work item forms, and other objects by using elements.	<ul> <li>Customize Project Tracking Data, Forms, Workflow, and Other Objects</li> <li>Determine your process and tracking customization requirements</li> </ul>
Look up the definition for a process template plug-in. You can create	

and customize your process templates, which you use to create team projects and standardize new processes.

• Customize Process Templates

## See Also

#### Concepts

Customize Team Projects and Processes Customize Process Templates <u>Configuring Your Server Using the Team Foundation Administration Console</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Using Team Foundation Server Command-Line Tools Send Feedback

You can perform many management and configuration tasks for Visual Studio Team Foundation Server at a command prompt. You can also use command-line tools to perform tasks for version control and build. As an alternative, you can use the administration console for Team Foundation to manage and configure your deployment of Team Foundation Server.

#### ⊠Note

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to use these commands on a server that is running Windows Server 2008. To open an elevated Command Prompt, click Start, right-click Command Prompt, and click Run as Administrator. For more information, see the <u>Microsoft Web site</u>.

# In This Section

#### Managing Server Configuration with TFSConfig

Describes the options and parameters of TFSConfig, a command-line tool that you can use to review and manage configuration settings for Team Foundation Server.

#### Configuring Lab Management with TFSLabConfig

Describes the options and parameters of TFSLabConfig, a command-line tool that you can use to manage the lab service provide by Visual Studio Lab Management.

#### **Changing Groups and Permissions with TFSSecurity**

Describes the options and parameters of TFSSecurity, a command-line tool that you can use to review and manage objects, application groups, users, and permissions for Visual Studio Application Lifecycle Management (ALM).

#### Specify the Bug Type to File By Using Microsoft Test Manager

By using the **tcm bugfieldmapping** command-line tool, you can import and export a mapping file to the team project. If you use this file, a tester can automatically create a work item for a code defect that was found by using Microsoft Test Manager.

#### Delete a Team Project [TFSDeleteProject]

Describes the options and parameters of TFSDeleteProject, a command-line tool that you can use to delete a team project.

Manage Project Server Integration with TfsAdmin ProjectServer

Describes the options and parameters of TfsAdmin ProjectServer, a command-line tool that you can use to configure the integration of TFS and Project Server.

<u>Upload or Download the Microsoft Project Mapping File [TFSFieldMapping]</u>

Describes the options and parameters of TFSFieldMapping, a commandline tool that you can use to upload or download the file that maps fields in Microsoft Project. By editing this file, you can customize how fields are defined for work items that are stored in Team Foundation and that are mapped to fields that are defined for Microsoft Project.

witAdmin: Customize and Manage Objects for Tracking Work Items

Describes the options and parameters of witAdmin, a command-line tool that you can use to extend the capabilities of Visual Studio ALM. You can also use this tool to customize your processes by modifying and administering objects that help you track work items.

# **Related Sections**

Configuring Your Server Using the Team Foundation Administration Console

Configuring Users, Groups, and Permissions

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Managing Server Configuration with TFSConfig
See Also Send Feedback

By using the **TFSConfig** command-line tool, you can configure or create a script to configure many aspects of your deployment of Visual Studio Team Foundation Server 2012 (TFS). For example, you can use the commands in **TFSConfig** to find information about or change the following items in esprtfs:

- service accounts and their passwords
- addresses and ports that TFS uses
- the state of team project collections that are part of the deployment
- the software updates that are applied to the application tier for Team Foundation and the databases for TFS
- information about other components, such as Team Foundation Build or Visual Studio Lab Management

#### ⊠Note

You can also configure a server interactively by using the administration console for Team Foundation. For more information, see <u>The Team</u> Foundation Administration Console.

You can use **TFSConfig** to configure TFS to use credentials, addresses, and ports that you have already defined. You cannot use the tool to create accounts or change the addresses that components such as SharePoint Products use. For example, you can use **TFSConfig** to change service accounts and passwords as long as the accounts and passwords already exist. You can also use **TFSConfig** 

to help repair or recover your deployment if an unexpected event occurs. For example, you can use it to repair your deployment if a software update does not apply to all components in the system or if a restoration of databases from a backup leaves a collection database improperly detached.

You must run **TFSConfig** from the application tier for Team Foundation. By default, **TFSConfig** is located in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to run **TFSConfig**. To open an elevated Command Prompt window, click Start, right-click Command Prompt, and click Run as Administrator. For more information, see this page on the Microsoft Web site: <u>User Account Control</u>.

To operate correctly, **TFSConfig** must be able to connect to the following servers and services:

- The instance or instances of SQL Server that host any databases that this deployment of TFS uses, such as the configuration database and databases for team project collections.
- Any SharePoint Web applications that this deployment of TFS uses.
- Any instance of SQL Server Reporting Services that this deployment of TFS uses.
- The Active Directory domain controller, if Team Foundation Server is deployed in a domain.
- Internet Information Services (IIS) on the application-tier server or servers.
- Any server that is running Team Foundation Build and is used by this deployment.
- Any server that is running Visual Studio Lab Management and is used by this deployment.

You can use the global options that are listed in the following table with most **TFSConfig** commands. The exceptions are noted in the specific topics for those commands.

Option	Description
/?	Displays the command syntax and options for <b>TFSConfig</b> .
/ <b>NoPrompt</b> or / <b>i</b>	Specifies that the administrator is not prompted before changes are committed.

# In This Section

Accounts Command

Authentication Command

**Certificates Command** 

**ChangeServerID** Command

**Collection Command** 

**ConfigureMail Command** 

**Diagnose Command** 

**Identities Command** 

Import Command (TFSConfig)

Jobs Command

Lab Commands

**License Command** 

Proxy Command (TFSConfig)

RebuildWarehouse Command

Recover Command

**RegisterDB** Command

**RemapDBs** Command

**Repair Command** 

Settings Command

Updates Command

# **See Also**

### **Other Resources**

Whitepaper: Unattended Installation (downloadable whitepaper)

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management Accounts Command
Example See Also Send Feedback

The **Accounts** command changes the accounts and passwords for all service accounts that Visual Studio Team Foundation Server (TFS) uses. These accounts include the service account for Team Foundation Server (TFSService), the data sources account for SQL Server Reporting Services (TFSReports), and the service account for Team Foundation Server Proxy (TFSProxy). You can also use this command to change the ownership of the databases that Team Foundation Server uses.

To change the user account or password interactively for the service account for Team Foundation Server (TFSService), you use the administration console for Team Foundation. For more information, see <u>Open the Team Foundation</u> <u>Administration Console</u>.

**Required Permissions** 

To use the **Accounts** command, you must be a member of the Team Foundation Administrators security group and a member of the sysadmin security group for any SQL Server databases that Team Foundation Server uses. If you use the **/proxy** option, you must be an administrator on the proxy server. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, click Start, right-click Command Prompt, and then click Run as Administrator. For more information, see this page on the Microsoft Web site: <u>User Account</u> Control.

TFSConfig Accounts /change|add|set|delete|updatepassword|resetowner

## **Parameters**

Argument	Description
AccountName	Specifies the name of the account that is used as the service account, in Domain\UserName or Computer\UserName format.
Password	Specifies the password for the account that is used as the service account.
ServerName	Used only with /resetowner /SQLInstance. Specifies the name of the server that hosts the databases for Team Foundation Server, and, if an instance other than the default instance is used, the name of the instance. If you specify an instance, you must use the following format: ServerName\InstanceName
	Used only with / <b>resetowner</b>

Database	Name / <b>DatabaseName</b> . Specifies the name of the database whose ownership will be changed.	
	Ontion	
	Option	
		C
		as
		or ad
		gr
/change		pe
		VC
		ĬA
		op
		ch
		А
		th
		ac
		gr
		pe
		ac
/add		W
		is T
		1 I ne
		SC
		ot
		se

	be en
	Se
	ас
	ad
	Tł
/set	op
	ha
	re
	or O
	SC
	Re
	ас
	Tł
	th
	ne
	th
/delete	10 DC
	pc wi
	is
	M
	th
	th
	cu
	ac
	U da
	ua Sr
	vr
	ус СО
/ResetOwner	th
1	

	th
	yc
	CC
	us
	or
	Cl
	ас
	ас
/UpdatePassword	pa
	yc
	Τŧ
	th
	SĮ
	yc
	us
	pe
	oţ
	CC
	(A
/ <b>AccountType:</b> {AdminConsole ApplicationTier ReportingDataSource Proxy}	ac Fc
	th
	R
	<b>(</b> T
	ас
	Se
	or
	Vð
	SĮ
	ас
	ch
	re
	/A

	Sŗ fo
	D
/Account:AccountName	C
	W
	su
	L
	C
	Fc
	to
	th
	to
	Sr
	se
	ß
/ <b>Password:</b> Password	T
	y
	01
	a
	S
	I I.
	Sr
	th
	th
	W
/SQLInstance:Serverivame	th
	m
	in
	Se

	SI
	Ua
/ <b>DatabaseName:</b> DatabaseName	W
	0V
	yt
	W
	$\mathbf{U}_{]}$
laantinua	av
/continue	CC
	us
	I I.
	ee an
	th
	A
	S
/usesqlalwayson	
	in
	Fc
	<u>A</u>
	<u>(S</u>

# **Remarks**

You use the **Accounts** command to automate changes to the service accounts, databases, and service account groups of Team Foundation Server. By using this command, you can configure accounts that you have already created, but you cannot create accounts.

Before you change the domain or workgroup of an account, the account must have the Account is sensitive and cannot be delegated permission on the application-tier server. For more information, see this page on the Microsoft Web site: <u>Enabling Delegated Authentication</u>.

# Example

The following example shows how to change the service account of data sources for Reporting Services to a new account in the Contoso domain, Contoso\NewAccount, and the password, to Password.

#### ⊠Note

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

#### Copy Code

TFSConfig Accounts /change /AccountType:ReportingDataSource /Account

The following example shows how to add the Network Service system account to the service account groups for Team Foundation Server. System accounts do not have passwords.

#### Copy Code

TFSConfig Accounts /add /AccountType:ApplicationTier /Account:"NT Au

The following example shows how to change the ownership of the database that is named TFS\_Warehouse on the server that is running ssnoversion and that is named "ContosoMain" on the named instance "TeamDatabases" to the user account under which you are running the command.

#### ĭNote

You cannot specify what account to set as the owner of the databases when you use this command. The owner will be set to the account under which you are running the command.

### Copy Code

TFSConfig Accounts /ResetOwner /SQLInstance:ContosoMain\TeamDatabase

# See Also

### **Other Resources**

<u>Using Team Foundation Server Command-Line Tools</u> <u>Team Foundation Server Services and Service Accounts</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Authentication Command
Example See Also Send Feedback

The **Authentication** command changes the network authentication protocol that a website for Team Foundation Server (TFS) uses.

**Required Permissions** 

To use the **Authentication** command, you must be a member of the Team Foundation Administrators security group on the application-tier server for Team Foundation If you use the **siteType** option, you must be an administrator on the application-tier server for Team Foundation or the proxy server. For more information, see <u>Team Foundation Server Permissions</u>.

#### ĭNote

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

TFSConfig Authentication [/provider:NTLM|Negotiate] [/viewAll] [/si1

### **Parameters**

Argumen	t Description
NTLM	Use with the <b>/provider</b> option to specify the NTLM authentication protocol.
Negotiate	Use with the / <b>provider</b> option to specify the Negotiate (Kerberos) authentication protocol.
Option	Description
/viewAll	Displays the current authentication settings for TFS.
/provider	Specifies the authentication provider you want to configure for the website.
	Specifies the website whose network authentication protocol you want to change.
/siteType	If you do not specify a name, the application tier for TFS is changed. If you specify
the proxy switch, Team Foundation Server Proxy is changed.

# **Remarks**

The **Authentication** command is used by an administrator who wants to change the network authentication protocol for one or more websites on which TFS relies. The administrator runs this command from the application tier to update those websites that require a change in their network authentication protocol. The command changes the **NTAuthenticationProviders** property in the IIS metabase.

#### **▲Important**

Before you use the **Authentication** command to change the authentication protocol, you should run the command with the **/viewAll** option to view the existing settings.

# Example

The following example displays the current value that is assigned for the network authentication protocol.

## Copy Code

>TFSConfig Authentication /viewAll

# See Also

## **Other Resources**

<u>Using Team Foundation Server Command-Line Tools</u> Administering Team Foundation Server <u>Team Foundation Server Services and Service Accounts</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Certificates Command
Example See Also Send Feedback

You can use the **Certificates** command to change how certificates are configured for client authentication in a deployment of Visual Studio Team Foundation Server (TFS) that utilizes HTTPS, secure sockets layer (SSL), and certificates. By default, the **Certificates** command will automatically select a client certificate from the certificate list for the current user. However, you can use the options for the command to specify a specific certificate or certificates from the current user context or from the local machine context.

Before you use the **Certificates** command, you must first configure the servers in your deployment of TFS to utilize certificates. For more information, see <u>Setting up HTTPS with Secure Sockets Layer (SSL) for Team Foundation</u> <u>Server</u>.

#### **Required Permissions**

To use the **Certificates** command, you must be a member of the Team Foundation Administrators security group and the local Administrators group on the computer from which you run the command. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

TFSConfig Certificates [/machine] [/disable] [/autoSelect] [/noprom

# **Parameters**

Placeholder	Desci	ription
Thumbprint	Specifies the hexadecimal string that contains the secure hash algorithm (SHA) in SHA-1 hash form of the certificate that you want to use. For more information about how to find the thumbprint for a certificate, see the following page on the Microsoft Web site: <u>How</u> to: <u>Retrieve the</u> <u>Thumbprint of a</u> <u>Certificate</u> .	
Ор	tion	Description
/machine		Specifies that the certificate list will be from the local machine context instead of the current user context.

/disable	Specifies that the client authentication certificate setting will be disabled.
/autoSelect	Specifies that a certificate will be automatically selected from the certificate list. The Manage Client Certificates window will not open.
/noprompt	Specifies that the Manage Client Certificates window will not open when the Certificates command is run.
	Specifies that the certificate that matches the specified thumbprint

	will be used.
/ <b>thumbprints:</b> thumbprint	You can
	specify more
	than one
	certificate by
	separating
	individual
	thumbprints
	with a
	comma.

# **Remarks**

You use the **Certificates** command to configure the client certificates that are used by a deployment of TFS that has been configured to use HTTPS/SSL and certificates. If you use the Certificates command with no options, a client certificate will be automatically selected from the current user context from which you run the command.

# Example

The following example shows how to specify the local machine certificate that has the thumbprint "aa bb cc dd ee" with no prompting.

#### ĭNote

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

TFSConfig Certificates /machine /thumbprint:aa bb cc dd ee /nopromp1

The following example shows how to specify using automatic selection of a client certificate from the current user store.

### Copy Code

TFSConfig Certificates /autoselect

# See Also

## Concepts

<u>Team Foundation Server Architecture</u> <u>Examples of Complex Topology</u>

### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
ChangeServerID Command
Example See Also Send Feedback

The **ChangeServerID** command changes the global universal identifications (GUIDs) that are associated with the databases for Visual Studio Team Foundation Server (TFS). GUIDs must be unique within a deployment of TFS. If more than one database has the same GUID, your deployment can become unstable or unusable. You can change the GUID for the configuration database, the GUIDs for all team project collection databases in the deployment, or both. Although you would not typically use this command in daily operations, you might use this command in the following circumstances:

- You restored your deployment to new hardware, the old deployment is still operational, and you want to utilize both deployments. This scenario is sometimes referred to as cloning the server.
- You want to test a software update or a hardware configuration on a duplicate deployment so that you do not risk disrupting your production environment.
- You want to fully test the restoration of databases to new hardware in a test lab or separate environment, to ensure that your deployment can be restored.
- You must reset the GUID for a collection database after moving it to another deployment for which that GUID is already reserved.

#### ⊠Note

The ChangeServerID command is not reversible. After a GUID has been changed, you cannot undo that change except by restoring a previous version of that database.

**Required Permissions** 

To use the **ChangeServerID** command, you must be a member of the Team Foundation Administrators security group and a member of the sysadmin security group for any SQL Server databases that Team Foundation Server uses. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

TFSConfig ChangeServerID /SQLInstance:ServerName] /DatabaseName:Con1

# **Parameters**

Placeholder	Descript	ion
ServerName	Specifies the name server that hosts configuration date TFS and the name instance if you we an instance other default instance. specify an instand must use the follow format:	ne of the the tabase for te of the vant to use than the If you ce, you owing
	ServerName\Inst	anceName
DatabaseName	Specifies the name configuration date default, the name database is TFS_Configuration	ne of the tabase. By e of this tonDB.
O	ption	Description
		Required. Specifies name of the server the

/SQLInstance:ServerName

pecifies the

server that is running SQL Server and the name of the instance if you want to use an instance other than the default instance. If you

	specify an instance, you must use the following format:
	ServerName\InstanceName
/ <b>DatabaseName:</b> DatabaseName	Required. Specifies the name of the configuration database for TFS.
/ProjectCollectionsOnly	Specifies that only the GUIDs for collections will be changed.
/ConfigDBOnly	Specifies that only the GUID for the configuration database will be changed.
/usesqlalwayson	Specifies that the databases are part of an AlwaysOn Availability Group in SQL Server. If configured, this option sets MultiSubnetFailover in the connection string. For more information, see <u>AlwaysOn Availability</u> <u>Groups (SQL Server)</u> .

# **Remarks**

You use the **ChangeServerID** command to create a discrete duplicate of a deployment of Team Foundation Server for testing or cloning purposes. After you use the ChangeServerID command, you must direct clients to create a connection to the changed server before it can be used.

# Example

The following example shows how to change the GUIDs of all databases in the Contoso1 deployment of TFS, where the configuration database is hosted on the server that is named "ContosoMain" on the named instance "TeamDatabases" in SQL Server.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

## Copy Code

TFSConfig ChangeServerID /SQLInstance:ContosoMain\TeamDatabases /Dat

# See Also

### Tasks

Move a Team Project Collection Split a Team Project Collection

### Concepts

Backing up and Restoring Your Deployment

#### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Collection Command
Example See Also Send Feedback

You can use the **Collection** command to attach, detach, or delete a team project collection from a deployment of Visual Studio Team Foundation Server (TFS). You can also use the **Collection** command to duplicate the database of an existing collection, rename it, and attach it to the deployment. This process is sometimes referred to as cloning a collection. However, you cannot use the **Collection** command to create a team project collection. For more information, see <u>Create a Team Project Collection</u>.

To manage collections interactively or to create a collection, you can use the Team Project Collections node in the administration console for Team Foundation. For more information, see <u>Open the Team Foundation</u> <u>Administration Console</u> and <u>Organizing Your Server with Team Project</u> <u>Collections</u>.

**Required Permissions** 

To use the **Collections** command, you must be a member of the following groups:

- the Team Foundation Administrators security group and the Administrators security group on the server or servers that are running the administration console for Team Foundation
- the sysadmin group on the server or servers that are running the instance of SQL Server that hosts the databases for Team Foundation Server
- the Farm Administrators group for the farm from which you are deleting the site collection, if your deployment uses SharePoint Products and you are using the /**delete** option.

For more information, see <u>Team Foundation Server Permissions</u>.

### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

TFSConfig Collection {/attach | /detach | /delete} [/collectionName:

# **Parameters**

Placeholder	Description
CollectionName	Specifies the name of the team project collection. If the name of the collection contains spaces, you must enclose the name in quotation marks (for example, "My Collection").
ServerName	Specifies the name of the server that hosts the configuration database for TFS, and the name of the instance if you want to use an instance other than the default instance. If you specify an instance, you must use the following format:
	ServerName\InstanceName
DatabaseName	Specifies the name of the configuration database. By default, the name of this database is TFS_ConfigurationDB.

Option	Description
/attach	Required if neither / <b>detach</b> nor / <b>delete</b> is used. If you specify this option, you must also use the / <b>collectionDB</b> option. As an option, you can also use / <b>collectionName</b> and / <b>clone</b> with this option. If you use the / <b>attach</b> option, the specified collection database will be added to your deployment of TFS.
/detach	Required if neither /attach nor /delete is used. If you specify this option, you must also use the /collectionName option. If you use the /detach option, the database for the

specified collection will be stopped, and the collection will be detached from your deployment of TFS.

Required if neither /**detach** nor /attach is used. If you specify this option, you must also use the /collectionName option. If you use the /**delete** option, the database for the specified collection will be stopped, and the collection will be permanently detached from TFS. You will not be able to reattach the collection database to this or any other deployment.

/delete

/CollectionName:CollectionName

The /**delete** option will not delete the collection database from SQL Server. After deleting the collection database from TFS, you can delete the database manually from SQL Server.

Required if either /**detach** or /**delete** is used. If you use this option with /detach or /**delete**, it specifies the collection that will be detached or deleted. If you use this option with /attach, it specifies a new name for the collection. If you use this option with both /attach and /clone, it specifies the name for the

	duplicated collection.
/ <b>CollectionDB:</b> ServerName;DatabaseName	Required if / <b>attach</b> is used. This option specifies the name of the server that is running SQL Server and the name of the collection database that is hosted on that server.
/clone	If you specify this option, the original collection database will be attached as a clone in SQL Server, and this database will be attached to TFS. This option is primarily used as part of splitting a team project collection.

# Example

The following example shows how to permanently remove the "Contoso Summer Intern Projects" team project collection from a deployment of Team Foundation Server.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

TFSConfig Collection /delete /CollectionName:"Contoso Summer Intern

TFSConfig - Team Foundation Server Configuration Tool Copyright © Microsoft Corporation. All rights reserved. Deleting a team project collection is an irreversible operation. A d Yes Found Collection 'Contoso Summer Intern Projects' Deleting... The delete of collection 'Contoso Summer Intern Projects' succeeded.

The following example shows how to duplicate the "Contoso Summer Interns Projects" team project collection, name it "Contoso Winter Interns Projects," and attach the duplicate collection to the deployment of Team Foundation Server.

TFSConfig Collection /attach /collectiondb:"ContosoMain;TFS\_Contoso

# See Also

### Tasks

Move a Team Project Collection Split a Team Project Collection

### Concepts

Backing up and Restoring Your Deployment

#### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
ConfigureMail Command
Example See Also Send Feedback

You can support the generation of automatic alerts when changes occur to work items and other conditions by configuring the mail server. The mail server that you specify supports all team projects defined for all project collections that are defined on the application-tier server where you run the command. For e-mail notifications to be sent to team members, you must configure the server that runs Team Foundation Server (TFS) to use an existing SMTP server by using the **ConfigureMail** command.

**Required Permissions** 

To use the **ConfigureMail** command, you must be a member of the Team Foundation Administrators security group on the Team Foundation applicationtier server. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function.

TFSConfig ConfigureMail /FromEmailAddress:emailAddress /SmtpHost:SMl

# **Parameters**

Option	Description
/FromEmailAddress:emailAddress	Specifies the address from which to send e-mail notifications from TFS for a check in, work item assigned to you, or other notifications. This address is also checked for validity and, depending on your server configuration, might have to represent a valid e-mail account on the mail server.
	If the address does not exist or is not valid, the default e-mail

address is used.

Specifies the name of the server that hosts the mail server.

/**SmtpHost:**SMTPHostName

# Example

The following example shows the syntax used to configure the from e-mail address to TFS@contoso.com and the SMTP mail server as ContosoMailServer:

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

## Copy Code

TFSConfig ConfigureMail /FromEmailAddress:TFS@contoso.com /SmtpHost:

# See Also

## **Other Resources**

Support alerts and feedback requests with an SMTP server Using Team Foundation Server Command-Line Tools

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Diagnose Command
Example See Also Send Feedback

You can use the **Diagnose** command to discover any software update mismatches between the application tier and the databases for Visual Studio Team Foundation Server (TFS). To operate correctly, the application tier and the databases must have the same software updates applied. If a software update mismatch occurs, you can use this command to identify the problem and receive information about what additional steps you must take to address the issues.

### ⊠Note

If the **Diagnose** command finds that a software update has been applied to the application tier but not to one or more databases for TFS, you can use the **Updates** command to attempt to automatically apply the update to the databases. For more information, see <u>Updates Command</u>. However, if a software update has been applied to one or more of the databases but not to the application tier, you cannot use the **Updates** command to apply that update to the application tier. The **Updates** command applies updates only to the databases. To address the problem of a missing update on the application tier, you must manually apply the update to the application tier.

#### **Required Permissions**

To use the **Diagnose** command, you must be a member of the Team Foundation Administrators group on the application-tier server for Team Foundation and a member of the sysadmin group in SQL Server on the data-tier server for Team Foundation. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

TFSConfig Diagnose /scope:updates

# **Parameters**

Option Description

/scope:updates /scope:updates

# **Remarks**

If the **Diagnose** command detects a mismatch, the command will provide information about the mismatch and suggest solutions to fix the problem.
# Example

The following example shows how to run the **Diagnose** command to determine whether the same software updates have been applied to the databases and the application tier.

### Copy Code

TFSConfig Diagnose /scope:updates

### See Also

### Reference

**Updates** Command

### Concepts

Team Foundation Server Architecture

### **Other Resources**

Upgrading Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Identities Command
See Also Send Feedback

The Identities command lists or changes the security identifier (SID) of users and groups in your deployment of Visual Studio Team Foundation Server 2012 (TFS). You might need to change or update the SID for users and groups in one of the following scenarios:

- changing the domain of your deployment
- changing from a workgroup to a domain or from a domain to a workgroup
- migrating accounts across domains in Active Directory

### ⊠Note

You do not need to run this command if you are changing domains within the same Active Directory forest. TFS will automatically handle SID changes for moves within the same forest.

### **Required Permissions**

To use the **Identities** command, you must be a member of the Team Foundation Administrators security group on the application-tier server for Team Foundation. You must also be a member of the sysadmin security group for SQL Server on the data-tier server for Team Foundation. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you are logged on with administrative credentials, you must open an

elevated Command Prompt to perform this function.

TFSConfig Identities [/change /fromdomain:DomainName1 /todomain:Domain

### **Parameters**

Placeholder	Description
DomainName	Specifies the name of the domain from which (/ <b>fromdomain</b> ) or to which (/ <b>todomain</b> ) you are changing SIDs. If you must specify a workgroup, use the computer name as the domain name.
AccountName	Specifies the name of the account that you want to list or change. When you use this parameter with / <b>change</b> , you specify the account from which you want to change after / <b>account</b> , and you specify the account to which you want to change after / <b>toaccount</b> . Do not include a domain or computer name with the account name.
	Specifies the name of the server that hosts the configuration database for TFS and the name of the

ServerName	instance if you want to use an instance other than the default instance. If you specify an instance, you must use the following format:	
	ServerName\Ins	stanceName
DatabaseName	Specifies the na configuration da e default, the nam database is TFS_Configura	me of the atabase. By le of this tionDB.
0	ption	Description
/change		Specifies that you want to change identities instead of listing them.
/fromdomain:	DomainName	Required when using / <b>change</b> . Specifies the original domain of the identities that you want to change. If you are changing from a workgroup environment, specifies the name of the computer.
		Required when using / <b>change</b> . Specifies the domain to which you want

/to	odomain:DomainName	to change identities. If you are changing to a workgroup environment, specifies the name of the computer.
/a	<b>ccount:</b> AccountName	Specifies the name of an account for which you want to list or change identities.
/to	oaccount:AccountName	Specifies the name of an account to which you want to change identities.
/S	<b>QLInstance:</b> ServerName	Specifies the name of the server that is running SQL Server and the name of the instance if you want to use an instance other than the default instance. If you specify an instance, you must use the following format: ServerName\InstanceName
/Ľ	DatabaseName:DatabaseName	Specifies the name of the configuration database for TFS.
		Specifies that the databases are part of an AlwaysOn Availability Group in SQL

/usesqlalwayson

Server. If configured, this option sets MultiSubnetFailover in the connection string.

For more information, see <u>AlwaysOn Availability</u> <u>Groups (SQL Server</u>).

# **Remarks**

You can optionally specify the database to change identities before you configure an application-tier server for the deployment. For example, you might specify the database to change the service account when you clone a deployment of TFS.

When you change identities, the target account or accounts must already exist in Windows.

You must wait for the next identity synchronization with Windows before the properties of accounts that you change with this command will be updated. This requirement includes changes from group to user, user to group, and domain account to local account.

# **Examples**

The following example shows how to list the names of all Windows users and groups that are stored in TFS and to display whether the SID for each user or group matches the SID in Windows. The Contoso1 domain administrators created domain groups such as "Contoso1\Developers" and "Contoso1\Testers" to help ease the management of permissions across TFS, SQL Server Reporting Services, and SharePoint Products.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

TFSConfig Identities

Sample output:

#### **₽C**0

TFSConfig - Team Foundation Server Configuration Tool Copyright © Microsoft Corporation. All rights reserved.

Account Name Exists (see note 1) Matches (see note 2)

CREATOR OWNER True True Contoso1\hholt True True BUILTIN\Administrators True True Contoso1\Developers True True Contoso1\Testers True True Contoso1\PMs True True Contoso1\jpeoples True True Contoso1\Domain Admins True True Contoso1\SVCACCT1 True True

9 security identifiers (SIDs) were found stored in Team Foundation 5

The following example shows how to change the SIDs for all accounts in Team Foundation Server from the Contoso1 domain to the SIDs for accounts that have matching names in the ContosoPrime domain. Only account names that match will have their SIDs updated. For example, if the "hholt" account exists as Contoso1\hholt and ContosoPrime\hholt, the account SID will be changed to the SID for ContosoPrime\hholt. If the "ContosoPrime\hholt" account does not exist, the SID will not be updated for Contoso1\hholt.

### Copy Code

TFSConfig Identities /change /fromdomain:Contoso1 /todomain:ContosoF

The following example shows how to change the account for a single user account, Contoso1\hholt, to the account for another user account, ContosoPrime\jpeoples.

### Copy Code

TFSConfig Identities /change /fromdomain:Contoso1 /todomain:ContosoF

The following example shows how to change the SID of the "NT AUTHORITY\NETWORK SERVICE" service account that is used in the deployment of Team Foundation Server when changing the domain of the deployment from Contoso1 to ContosoPrime. To change a system account such as Network Service, you must follow a two-stage process. You first change the service account from NT AUTHORITY\NETWORK SERVICE to a domain account in the new domain (TempSVC), and then you change the account back to NETWORK SERVICE on the server in the new domain. The configuration database is hosted on the server that is named "ContosoMain" on the named instance "TeamDatabases" in SQL Server.

TFSConfig Identities /change /fromdomain:"NT AUTHORITY" /todomain:Cc TFSConfig Identities /change /fromdomain:ContosoPrime /todomain:"NT

## See Also

### **Other Resources**

<u>Using Team Foundation Server Command-Line Tools</u> Administering Team Foundation Server <u>Team Foundation Server Services and Service Accounts</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Import Command (TFSConfig)
Example See Also Send Feedback

You can use the **Import** command to import the data and projects from Visual Studio Team System 2008 Team Foundation Server into a new team project collection. To use this command, the databases that you import must be hosted on a server that is running SQL Server 2008 R2. If your databases are hosted on a server that is running SQL Server 2005, you must first upgrade them to a supported version of SQL Server before you can use the **Import** command. If you are aggregating multiple servers into a single deployment of TFS, you must run the Import command against each server separately. The projects from each server on which you run the Import command will be incorporated into a single team project collection. For example, if you run the Import command on two separate servers, you will create two team project collections. Each collection will contain all the projects and data from one server.

The **Import** command irrevocably changes the databases to function with the current version of TFS. After you run the **Import** command, you will no longer be able to use the databases with previous versions of TFS. After you import the databases, users of the projects and data that are contained in those databases must create a connection to the collection that contains the projects that they want to access before they can work in those projects. For more information, see Connect to Team Projects in Team Foundation Server. If the command fails or you want to roll back the projects to an earlier version of TFS, the only way to restore the projects to the earlier version is to restore the databases from a backup that was made before you used the **Import** command.

### **▲Important**

Make sure that you back up all databases before you actually use the **Import** command.

The **Import** command is not intended to be a substitute or replacement for the upgrade wizard for TFS. This command does not upgrade reports or team project portals that are associated with the projects and databases to TFS. The primary use for this command is to aggregate multiple deployments of previous versions of TFS into a current deployment of TFS. In most cases, you should first upgrade a deployment of TFS by using the upgrade wizard, and you should then move or split projects into collections after the upgrade is completed. This approach will upgrade reports along with the deployments and help preserve the functionality of reports and team project portals for the deployment. For the most recent information about how to upgrade previous versions of Team Foundation. You can find the most recent version of this guide on the following page on the Microsoft Web site: Installing Team Foundation Server.

### **Required Permissions**

To use the **Import** command, you must be a member of the Team Foundation Administrators group on the application-tier server for Team Foundation. You must also be a member of the sysadmin group in SQL Server on the data-tier server for Team Foundation. For more information, see <u>Team Foundation Server</u> <u>Permissions</u>.

#### ⊠Note

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

TFSConfig Import /SQLInstance:ServerName /CollectionName:NameOfNewCollectionName

### **Parameters**

Argument	Description
	Specifies the name of the server that hosts the databases. If you use an instance other than the default instance, you must also specify its name in the following format:
	ServerName\InstanceName
ServerName	The name of the server and instance that is specified must be an actual computer name, IP address, or system name such as "localhost." The command will fail if you use a placeholder, such as "." or "(local)".
NameOfNewCollection	Specifies the name of the team project collection that you will create as part of importing the databases. If the name of the collection that you want to create contains spaces, you must enclose the name in quotation marks (for example, "My

Collection").	
Option	Description
/ <b>SQLInstance:</b> ServerName	Specifies the server that hosts the databases that you want to import into a team project collection.
/ <b>CollectionName:</b> NameOfNewCollection	Specifies the name of the collection that will host the imported databases.
/confirmed	Confirms that you have backed up all the databases before you run the command.

# Example

The following example shows how to import the databases in the Contoso1 deployment of TFS. In this example, the databases are hosted on the 2008Projects instance on the server that is running SQL Server and that is named DataServer2 (DataServer2\2008Projects). The projects will all be imported into the team project collection that is named "Old Server."

### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

TFSConfig Import /SQLInstance:DataServer2\2008Projects /CollectionNa

### **See Also**

### Tasks

<u>Create a Team Project Collection</u> <u>Add a SharePoint Web Application to Your Deployment</u> <u>Add a SharePoint Web Application to a Team Project Collection</u> <u>Add a Report Server to Your Deployment</u> <u>Add a Report Server to a Team Project Collection</u>

### **Other Resources**

Upgrading Team Foundation Server Add a Team Project Portal Add Reports to a Team Project

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Jobs Command
Example See Also Send Feedback

You can use the **Jobs** command to create a log file that provides the details of the most recent job activity for a specific team project collection. Jobs are run at any time when an update is required to change the information or configuration for a collection. You can also use this command to retry a job for one or all team project collections.

To retry a job interactively, you can open the administration console for Team Foundation, click the Status tab for the collection, and then click Retry Job. For more information, see <u>Open the Team Foundation Administration Console</u>.

**Required Permissions** 

To use the **Jobs** command, you must be a member of the Team Foundation Administrators security group. For more information, see <u>Team Foundation</u> <u>Server Permissions</u>.

#### ĭNote

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

```
TFSConfig Jobs /retry [/CollectionName:CollectionName] [/Collection]
TFSConfig Jobs /dumplog [/CollectionName:CollectionName] [/Collectic
```

## **Parameters**

Placeholder	Description	
CollectionName	Specifies the name the team project collection. If the n of the collection contains spaces, ye must enclose the n with quotation ma (for example, "My Collection") If you want to specify all collections, you ca use an asterisk (*)	e of ame ou ame rks 7 1
ID	Specifies the identification num of the team projec collection.	ber t
Op	otion	Description
		Required if / <b>dumplog</b> is not used. Specifies that the most recent job will be reattempted for the specified

/retry	team project collection. If you use this option, you must also use either the / <b>CollectionName</b> or the / <b>CollectionID</b> option.
/dumplog	Required if / <b>retry</b> is not used. Specifies that the most recent job activity for the collection will be sent to a log file. If you use this option, you must also use either the / <b>CollectionName</b> or / <b>CollectionID</b> option.
/ <b>CollectionName:</b> CollectionName	Required if / <b>CollectionID</b> is not used. Specifies the name of the collection for which the most recent job will be either retried (/ <b>retry</b> ) or logged (/ <b>dumplog</b> ). If

you want to specify all collections, you can use an asterisk (\*).

Required if /CollectionName is not used. Specifies the identification number of the collection for which the most recent job will be either retried (/**retry**) or logged (/dumplog).

### /CollectionID:ID

# Example

The following example shows how to create a log file that lists the most recent job activity for the "Contoso Summer Intern Projects" team project collection in Team Foundation Server.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

TFSConfig Jobs /dumplog /CollectionName:"Contoso Summer Intern Proje

### See Also

### Tasks

Move a Team Project Collection Split a Team Project Collection

### Concepts

Backing up and Restoring Your Deployment

### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Lab Commands
See Also Send Feedback

Visual Studio Lab Management enables you to manage virtual environments used in the development, deployment, and testing of your software applications. Lab Management works with Microsoft System Center Virtual Machine Manager (SCVMM) to accomplish these tasks.

You can use the **TfsConfig Lab** commands to configure Lab Management in the Team Foundation Server Application Tier and to add SCVMM host groups and library shares to team project collections.

- A host group is a logical container that is used to manage one or more physical machines that host virtual machines. Host groups are created by an administrator in SCVMM.
- A library share is a designated share that provides access to the file-based resources such as ISO images and virtual hard disks that Lab Management uses to create and deploy virtual environments. Library shares are created in SCVMM by the SCVMM administrator.

To manage the host groups and library of individual team project from the command line, use the TfsLabConfig tool. For more information, see <u>Configuring Lab Management with TFSLabConfig</u>.

### **Common Tasks**

Task

Related Content

Set up Team Lab Management and configure network isolation. The **Settings** options identify the SCVMM that will be used by Lab Management and the network location that will be used for virtual machines. Other options TFSConfig enable you to configure Lab /Settings Network Isolation which **Commands** allows you to run multiple copies of a lab environment to run at the same time without causing network conflicts, such as conflicts in computer names and Domain Name System (DNS) registration.

Manage SCVMM host groups for a team project collection. The **HostGroup** options lets you assign a host group<br/>that was created inTFSConfig<br/>LabSCVMM to a team<br/>project collection,/HostGroup<br/>Commandsremove an assignment,<br/>or modify LabManagement properties<br/>of the host group.

Manage SCVMM library shares for a team project collection. The **LibraryShare** options lets you assign a library share to a team project collection, remove an assignment, or modify Lab Management properties of the library share.

TFSConfig Lab /LibraryShare Commands

Reassign DNS records to moved or modified team project collections. When you move a team project collection from one instance of Team Foundation Server to another, or when you change the Team Foundation Server service account, you might have to reassign DNS names that were generated by Lab Management. The **DNS** options enable this

TFSConfig Lab /DNS Commands procedure.

### See Also

### Concepts

<u>Configuring Your Server Using the Team Foundation Administration Console</u> <u>Configuring Lab Management with TFSLabConfig</u>

### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSConfig Lab /Settings Commands
Example See Also Send Feedback

You can configure Visual Studio Lab Management by using the **TFSConfig Lab** /**Settings** option. The Settings option

- Sets the name of the System Center Virtual Machine Manager (SCVMM) Server that controls the administration of virtual machines in your lab.
- Sets the network location, such as the network domain or workgroup, that the physical computers in all host groups can connect to.
- Sets the IP addresses and virtual DNS suffix for network-isolated networks in your lab.

```
TfsConfig Lab
/Settings
[/ScVmmServerName:VMMServerName]
[/NetworkLocation:networkLocation]
[/IpBlock:networkIsolationIpRange]
[/DnsSuffix:networkIsolationDnsSuffix]
[/NoPrompt]
[/List]
```

# Options

Option	Description
ScvmmServerName:VMMServerName	Optional. Sets the fully qualified name of the System Center Virtual Machine Manager 2008 (SCVMM) server that will be used by Team Foundation Server. This is the name of the SCVMM server that will be used to manage the virtual machines. For Team Foundation Server to communicate with SCVMM, you must add the account under which Team Foundation Server is running to the Administrator role in SCVMM.
	Optional. Sets the

#### NetworkLocation:networkLocation

fully qualified name of a network, such as a network domain or workgroup, that is available on all the hosts in your lab network. When it provisions a virtual machine, Lab Management automatically connects the virtual machine to the specified network. You can find the available network locations on a host using **SCVMM** Administrator Console.

Optional. Sets the range of IP addresses to be assigned to the virtual machines in an environment when an isolated network is created. Because the IP addresses are used only for internal routing among virtual machines and are not exposed beyond IpBlock:networkIsolationIpRange

the boundaries of an environment, you can specify any IP range that is not used within the network specified by the /NetworkLocation option. In most cases, the default range of 192.168.23.0/24 should work. If you encounter problems connecting to network isolated environments, you might have to choose a different range.

Optional. Sets the DNS suffix that will be used to register the names of virtual machines on the isolated network for the virtual environment. To confirm that the suffix is configured correctly in the DNS hierarchy, contact your network

**DnsSuffix:**networkIsolationDnsSuffix

	administrator.
NoPrompt	Optional. Do not prompt for confirmation. Do not use with the / <b>List</b> option.
List	Lists the current configuration values for Lab Management.

# **Remarks**

One of the /ScvmmServerName, /NetworkLocation, /IpBlock, /DnsSuffix, or /List options must be specified on each TfsConfig Lab /Settings command line.

To set up Lab Management, you must specify both /**ScVmmServerName** and the /**NetworkLocation** options. However, you can specify these options on separate command lines.

To set up network isolation, you must specify both the /**IpBlock** and /**DnsSuffix** options. However, you can specify these options on separate command lines.

Network isolation enables you to use multiple copies of a virtual machine without machine name or IP address conflicts. You must assign both a DNS suffix and an IP range for an isolated network. Because the IP addresses are used only for internal routing among virtual machines and are not exposed beyond the boundaries of an environment, you can specify any IP range that is not used within your public network. In most cases, the default range of 192.168.1.0/24 should work. If you encounter problems connecting to network-isolated environments, you might have to choose a different range.

# Example

In the first example, Lab Management is set up by using the /ScvmmServerName and /NetworkLocation options on a single command line. In the second example, network isolation is configured by using the /IpBlock and /DNSSuffix options on separate command lines.

### Copy Code

REM First example tfsconfig lab /settings /scvmmservername:vmmserver /networklocation: REM Second example

tfsconfig lab /settings /ipblock: 192.168.23.0/24 tfsconfig lab /settings /dnssuffix:virtual1.lab1.contoso.com
## **See Also**

### Reference

TFSConfig Lab /LibraryShare Commands TFSConfig Lab /HostGroup Commands

### Concepts

TFSConfig Lab Command Reference TFSLabConfig Command Reference <u>Configuring and Administering Lab Management</u>

#### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSConfig Lab /HostGroup Commands
Example See Also Send Feedback

You can use the **TfsConfig Lab** /**HostGroup** commands to add, edit, or delete the assignment of a System Center Virtual Machine Manager (SCVMM) host group to a team project collection. Host groups that are assigned in this manner are managed by Visual Studio Lab Management.

```
TfsConfig Lab /hostgroup /CollectionName:collectionName
      { /Add
            /SCVMMHostGroup:vmmHostPath
            /Name:name
            [LabEnvironmentPlacementPolicy:{Conservative|Aggressive]
            [/AutoProvision:{True|False}]
            [/DNSSuffix:dnsSuffix]
       | /Delete
            /Name:name
            [/NoPrompt]
       | /Edit
            /Name:name
            {[/AutoProvision:{True|False}]
            [/LabEnvironmentPlacementPolicy:{Conservative|Aggressive
            [/DNSSuffix:dnsSuffix]}
            [/NoPrompt]]
       | /List
       / /ListVmmHostGroups }
```

## **Parameters**

	Option	De
<b>CollectionName:</b> collecti	onName	Required. The project collecti application-tien Server.
Add		Adds the special group to the tea host groups. Yo / <b>SCVmmHost</b> options with <b>A</b>
Delete		Removes the sj from the team j You must speci with <b>Delete</b> .
		Sets one or bot Management A LabEnvironm properties for t
Edit		You must speci and at least one / <b>AutoProvisio</b> / <b>LabEnvironn</b> options with <b>E</b>

SCVMMHostGroup:vmmHostGroupPath	Required with Specifies the h SCVMM host
Name:name	Required with / <b>Edit</b> options. S the team project group to add, d
AutoProvision: {True False}	Optional with t options. Sets ('I ( <b>False</b> ) the Aut of the host grou specifies wheth automatically <i>a</i> project in the c host groups are projects in a co use the <b>TfsCor</b> command.
	Optional with t options. Specif Management tu machines in a l it deploys new environments.
LabEnvironmentPlacementPolicy:{Conservative Aggressive	<ul> <li>Conserva Consider environme decisions.</li> <li>virtual ma of enviror</li> <li>"Stopped"</li> </ul>

• Aggressiv non-runni environme decisions.

Optional. Sets the virtual com group.

- If the /**D**N specified ' value, sets suffix the computers host comp group.
- If the /**D**N specified ' the suffix are set to 1 their host group.
- If the /**D**N specified ' the suffix is not chai

Optional with t options. Do no confirmation.

Displays the hc the team projec

### DNSSuffix:[dnsSuffix]

### NoPrompt

List

### ListVmmHostGroups

Displays the harmonic from SCVMM

# **Remarks**

Host groups are containers that an administrator creates within SCVMM to group a set of virtual machine hosts for easy management. Host groups are hierarchical; a host group can contain other host groups.

Each host group is identified by its host path, a sequence of host group names that specifies the location of a host or host group within the hierarchy of host groups in SCVMM. All host paths begin with the root host group. For example, the host path All Hosts\New York\Site21\VMHost05 indicates that the host VMHost05 belongs to the host group Site21, which is a child host group of the host group New York.

Use only one of the /**Add**, /**Delete**, or /**Edit** options on a command line. Use separate **TfsConfig Lab** /**HostGroup** command lines to assign multiple host groups to a team project collection.

You can also use **TfsConfig Lab** /**HostGroup** commands to set properties that are specific to Lab Management:

- AutoProvision specifies whether the host group is assigned to each team project in the team project collection. By default, AutoProvision is on. To assign a host group in a project collection to an individual team project, use the <u>TFSLabConfig CreateTeamProjectHostGroup Command</u>.
  - **True** (Default). The host group is assigned to each team project in the team project collection.
  - **False**. The host group is not assigned to each team project in the team project collection.
- LabEnvironmentPlacementPolicy specifies whether Lab Management considers the existing virtual machines when it deploys new environments on a physical machine in a host group.
  - **Conservative** (Default). Consider non-running virtual environments in deployment decisions. This includes all virtual machines that are part

of environments and that are in "Stopped" state also.

- **Aggressive** Do not consider non-running virtual environments in deployment decisions.
- **DNSSuffix** specifies a DNS suffix to use for the virtual computers created in the host group. The following table describes how the DNS suffixes of virtual computers are affected by the /DNSSuffix setting.

DNSSuffix	/Add	/Edit
<b>DNSSuffix:</b> dnsValue	The DNS suffix is set to dnsValue.	The DNS suffix is set to dnsValue.
DNSSuffix:	The DNS suffix is inherited from host computer.	The existing suffix value is removed and the DNS suffix is inherited from host computer.
<not specified=""></not>	The DNS suffix is inherited from host computer.	DNS suffix not changed.

# Example

In the following example a SCVMM host group is assigned to a team project collection. Because the /**AutoProvision** options is not specified, the host group is automatically assigned to all team projects in the collection.

tfsconfig lab /hostgroup /add /scvmmhostgroup:"All Hosts\Lab1\HostGr

## **See Also**

### Reference

TFSConfig Lab /Settings Commands TFSConfig Lab /LibraryShare Commands TFSConfig Lab /HostGroup Commands

### Concepts

TFSConfig Lab Command Reference TFSLabConfig Command Reference <u>Configuring and Administering Lab Management</u>

#### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSConfig Lab /LibraryShare Commands
See Also Send Feedback

You can use the **TfsConfig Lab** /**LibraryShare** command to add, remove, or edit the assignment of a library share from System Center Virtual Machine Manager (SCVMM) to a team project collection. You can also use this command set properties that are specific to Visual Studio Lab Management and to display the library shares that are currently assigned to a collection in Lab Management or to display all the library shares in SCVMM.

```
TfsConfig Lab /LibraryShare
    /CollectionName:collectionName
    { /Add
        /SCVMMLibraryShare:librarySharePath
        /Name:name
         [/AutoProvision:{True|False}]
    | /Delete
        /Name:name
        [/NoPrompt]
    | /Edit
        /Name:name
        /AutoProvision:{True|False}
        [/NoPrompt]
    | /List
    | /List
    | /ListSCVMMLibraryShares }
```

## **Parameters**

Option	Description
Add	Adds the specified library share to the team project collection. You must specify the /SCVMMLibraryShare and /Name options with Add.
Delete	Removes the specified library share from the team project collection. You must specify the / <b>Name</b> option with <b>Delete</b> .
	Sets or clears the AutoProvision property of the library share. You must specify the / <b>Name</b> and / <b>AutoProvision</b> options with <b>Edit</b> .
Edit	By default, library shares are assigned to the team projects in a collection.

Changing autoprovision does impact existing team projects.

•

Required. Specify the name of the team project collection on the application-tier Team Foundation Server.

SCVMMLibraryShare:librarysharePath Required with Add. Specifies the path to the Virtual Machine Manager library share.

Name:libraryShareName

CollectionName: collectionName

Required with **Add**, **Delete**, and **Edit**. Specifies the name of library share in the team project collection.

Optional with **Add**; required with **Edit**. Specifies whether the library shares are automatically assigned to each team project in the collection. By default, library shares are assigned to the team projects.

#### **AutoProvision**

NoPrompt	Optional with <b>Add</b> and <b>Edit</b> . Do not prompt the user for confirmation.
List	Lists all library shares that are assigned to the specified team project collection.
ListSCVMMLibraryShares	Lists all library shares that are available in Virtual Machine Manager.

# **Remarks**

A library share is a designated share on a Virtual Machine Manager library server. A library share provides access to file-based resources for virtual Lab Management environments that are stored on your library servers, such as ISO images and virtual hard disks. Library shares are created in Virtual Machine Manager. Visual Studio Lab Management uses library shares to provision virtual machines in the lab.

Use only one of the /**Add**, /**Edit**, or /**Delete** options on a TfsConfig Lab /LibraryShare command line. Use separate **TfsConfig Lab** /**LibraryShare** command lines to assign multiple library shares to a collection.

By default, library shares in a team project collection are automatically assigned to each of the team projects in the collection. You can use the **/AutoProvision** option with the **/Add** and **/Edit** commands to change the library shares are assigned.

- Set the /**AutoProvision** option to **False** to disable the automatic assignment of the library share to team projects. To assign or remove a library share in an individual team project, use the **TfsLabConfig** <u>TFSLabConfig</u> <u>CreateTeamProjectLibraryShare Command</u>.
- Set the /AutoProvision option to True to enable the automatic assignments.

## **See Also**

### Reference

TFSConfig Lab /Settings Commands TFSConfig Lab /HostGroup Commands

### Concepts

TFSConfig Lab Command Reference TFSLabConfig Command Reference <u>Configuring and Administering Lab Management</u>

#### **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSConfig Lab /DNS Commands
Example See Also Send Feedback

The **TfsConfig Lab** /**DNS** command adds, deletes, or displays DNS records that were created by Visual Studio Lab Management for network-isolated environments.

Team Foundation Server uses the suffix you enter when it registers a unique external name with DNS for each virtual machine in a network-isolated environment. The DNS alias record makes it possible for machines and other objects outside the isolated network to communicate with machines inside the isolated network. Because Team Foundation Server goes into the DNS zone to register the alias record, the service account under which Team Foundation runs must have permissions to add or delete alias records in the specified DNS zone.

If your Team Foundation Server deployment has more than one application tier and each application tier runs under a different service account, then each application-tier service account must have permission to edit the DNS alias records created by the other application tiers.

Note DNS record management is performed automatically by Lab Management. You should use the **/DNS** command only in the following situations:

- You change the account under which Visual Studio Team Foundation Server (TFS) runs.
- You move a team project collection from one instance of Team Foundation Server to another.

In both these cases, the DNS records that were created by using the old TFS service account have to be removed, and then the same DNS records have to be

re-created by using the new TFS service account. If these steps are not performed in the previous scenarios, the new TFS service account will not be able to perform automatic management of those DNS records. As a result, users will not be able connect to virtual environments.

```
TfsConfig Lab /DNS
{/Add | /Delete | /List}
  [/CollectionName:collectionName |
   / CollectionName:collectionName /TeamProject:projectName |
   / CollectionName:collectionName /TeamProject:projectName /LabEnv
   /Name:FQDN /IP:IpAddress]
   [/NoPrompt]
```

# Options

Option	Description
Add	Adds the specified DNS records. To use the / <b>Add</b> option, the targeted environments must be running.
Delete	Removes the specified DNS records.
List	Displays the specified DNS records.
LabEnvironment:environmentUri	Targets the / <b>Add</b> , / <b>Delete</b> , or / <b>List</b> options to an individual network-isolated environment that is specified by environmentUri.
	To use the LabEnvironment option, you must also specify the

/**Collection** and /**TeamProject** options.

When used without /LabEnvironment, targets the /Add, /Delete, or /List options to all of the network-isolated environments in the team project that is specified by projectName. Otherwise, /TeamProject specifies the team project that contains the environment.

To use the /**TeamProject** option, you must also specify the /**Collection** option

When used without /TeamProject, targets the /Add, /Delete, or /List options to all of the network-isolated environments in the team project collection that is specified by

TeamProject:projectName

CollectionName:collectionName

	collectionName. Otherwise, / <b>Collection</b> specifies the team project collection that contains the team project.
Name:FQDN	Specifies the fully qualified domain name of the network location that contains the environment to target.
	You must specify the / <b>Name</b> and / <b>IP</b> options together.
<b>IP:</b> IPAddress	Specifies the IP address of the environment to target.
	You must specify the / <b>Name</b> and / <b>IP</b> options together.

# **Remarks**

Specify only one of the /Add, /Delete, or /List options on a TfsConfig Lab /DNS command line. If you do not specify any target options, the operation acts on all virtual machines of all network isolated environments that belong to all project collections in the Team Foundation Server database.

To target the DNS entries of network isolated environments of an object in the Lab Management object hierarchy, use the appropriate combination of **/Collection**, **/TeamProject**, and **/LabEnvironment** options

 The /LabEnvironment option targets the command to the specified network-isolated environment. You must use the /CollectionName and /TeamProject options with the /LabEnvironment option to specify the team project collection and the team project that contain the environment.

Use the format

**vstfs:///LabManagement/LabEnvironment**/environmentID to specify the environment URI. You can view the environment identifier (environmnetID) in the Environment Viewer of Lab Management or from the description of the virtual machine in SCVMM Administrator Console.

- The /**TeamProject** option targets the operation to isolated network environments in the specified team project. The /**TeamProject** option must be used with the /**CollectionName** option and the /**CollectionName** option must specify the team project collection that contains the team project.
- The /**CollectionName** option targets the operation to network isolated environments in the specified team project collection.

The second way to target a network isolated environment is to use the /**Name** and /**IP** options to specify the fully qualified external name and external IP address of an individual virtual machine. You must specify both the /**Name** and /**IP** options on the command line. You can view the external name and external IP address of a virtual machine in the Environment Viewer of Lab Management or from the description of the virtual machine in SCVMM Administrator Console.

# Example

In the first example, records for all network-isolated environments in a team project are added to DNS. In the second example, an individual DNS record is removed.

Copy Code

REM First example tfsconfig lab /dns /add /collectionname:Collection0 /teamproject:Prc REM Second example tfsconfig lab /dns /delete /name:0b668996-2736-46d2-88ac-0733acbd0d{

## See Also

### Concepts

TFSConfig Lab Command Reference TFSLabConfig Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSConfig Lab /Delete Command
Example See Also Send Feedback

Use the **TfsConfig Lab** /**Delete** option to remove all group hosts, library shares, and environments from a specified team project collection. By default, the associated objects in the System Center Virtual Machine Manager (SCVMM) are not deleted. You can add the /**External** option to the command line to remove the objects from the project collection and from SCVMM.

#### ⊠Note

The **/Delete** option works on all lab assets in a team project collection only when the **/LibraryShare** and **/GroupHost** options are not specified on the command line. For more information, see <u>TFSConfig Lab /LibraryShare</u> <u>Commands</u> and <u>TFSConfig Lab /HostGroup Commands</u>.

TfsConfig Lab /Delete
 /CollectionName:collectionName
 [/External]
 [/NoPrompt]

# Options

Option

Required. Specifies the name of the team project CollectionName:collectionName collection on the application tier of Team Foundation Server.

> Optional. When specified, removes the library shares and host groups in SCVMM in addition to the objects in the team project collection. When /External is not

Description

External

specified, the **TfsConfig Lab /Delete** command only removes the objects in the team project collection.

NoPrompt

Optional. When specified, does not display progress and success information.

## **Remarks**

Use the **/Delete** command to remove the lab assets from a project collection when you want to detach the lab configuration of a project collection. This is useful when moving a project collection from one Team Foundation Server (TFS) instance to another, and where the new Team Foundation Server instance uses a different SCVMM server than the original one. In this case, you will have to delete all the lab assets and reconfigure lab for the project collection.

# Example

In the following example, all lab objects in the DefaultCollection team project collection are removed. Because the **/External** option is not specified, the objects are retained in SCVMM.

### Copy Code

tfsconfig lab /delete /collectionName:DefaultCollection

## See Also

### Concepts

TFSConfig Lab Command Reference TFSLabConfig Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
License Command
Example See Also Send Feedback

You can use the **License** command to display, modify, or extend the licensing key for your deployment of Visual Studio Team Foundation Server.

To view, modify, or change the licensing for your deployment interactively, you can use the administration console for Team Foundation. For more information, see <u>Open the Team Foundation Administration Console</u> and Locate or Change the Product Key for Team Foundation Server.

**Required Permissions** 

To use the **License** command, you must be a member of the Team Foundation Administrators security group. For more information, see <u>Team Foundation</u> <u>Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, click Start, right-click Command Prompt, and then click Run as Administrator. For more information, see this page on the Microsoft Web site: <u>User Account Control</u>.

TFSConfig License [/ProductKey:Key] [/extend [NewTrialID]]

## **Parameters**

Placeholder	Description	
Key	Specifies a valid product key to use when changing your deployment from Team Foundation Server from a trial license to a full license.	
NewTrialID	Specifies a second trial license key. A second trial license key can be obtained from Microsoft Corporation. The primary trial license key is good for 90 days, but the secondary trial license key will extend your deployment for only 30 days.	
Option	Description	
/ProductKe	Specifies that the license key for the y:Key deployment will be updated with the value of Key.	

	Specifies that the
	trial licensing
	period for Team
	Foundation Server
	will be extended by
	30 days. This option
/extend	can be used only
	once. If a second
	extension is
	required, you must
	obtain a second trial
	license from
	Microsoft.

# Example

The following example shows how to view the licensing information for a deployment of Team Foundation Server. In this example, the deployment is using a trial license.

#### ⊠Note

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

#### Copy Code

TFSConfig License

TFSConfig - Team Foundation Server Configuration Tool Copyright © Microsoft Corporation. All rights reserved. Team Foundation Server Standard license The following features are installed: Team Foundation Server Build Services Build: 21106.00 Product ID: 01234-567-8910 Trial license with 74 days remaining, expiring on 6/30/2010 Trial ID: ABCD-EFGH-IJKL

The following example shows how to extend the trial license for a deployment of Team Foundation Server. In this example, the trial from the previous example is extended for 30 days.

#### Copy Code

TFSConfig License /extend

TFSConfig - Team Foundation Server Configuration Tool Copyright © Microsoft Corporation. All rights reserved. Now extending trial Team Foundation Server Standard license The following features are installed: Team Foundation Server Build Services Build: 21106.00 Product ID: 01234-567-7654987 Trial license with 134 days remaining, expiring on 8/30/2010 Trial ID: XYZZY-REDFG-WESDF

The following example shows how to update the trial license to a full license.

#### Copy Code

TFSConfig License /ProductKey:abcde-fghij-klmno-pqrst-uvwxy

TFSConfig - Team Foundation Server Configuration Tool Copyright © Microsoft Corporation. All rights reserved. Now applying Product Key 'abcde-fghij-klmno-pqrst-uvwxy' Team Foundation Server Standard license The following features are installed: Team Foundation Server Build Services Build: 21106.00 Product ID: 01234-567-765456 Retail license

## See Also

### Tasks

Locate or Change the Product Key for Team Foundation Server

### Concepts

Administering Team Foundation Server

### **Other Resources**

Upgrading Team Foundation Server

Send <u>feedback</u> on this topic to Microsoft.
Visual Basic C#
Visual C++
F#
HLSL
JScript
Proxy Command (TFSConfig)
Example See Also Send Feedback

You can use the **TFSConfig Proxy** command to update or change the settings used by Team Foundation Server Proxy. Team Foundation Server Proxy provides support for distributed teams to use version control by managing a cache of downloaded version control files in the location of the distributed team. By configuring Team Foundation Server Proxy, you can significantly reduce the bandwidth needed across wide area connections. In addition, you do not have to manage downloading and caching of version files; management of the files is transparent to the developer who is using the files. Meanwhile, any metadata exchanges and file uploads continue to appear in Team Foundation Server (TFS). If you use the Team Foundation Service to host your development project in the cloud, you can use the Proxy command to not only manage the cache for projects in the hosted collection, but also to manage some of the settings used by that service.

For more information about installing Team Foundation Server Proxy and initial configuration of the proxy, see How to: Install Team Foundation Proxy and Set Up a Remote Site. For more information about configuring proxy on client computers, see <u>Team Foundation Version Control Command Reference</u>.

### Requirements

To use the **Proxy** command, you must be a member of the Team Foundation Administrators security group and an administrator on the proxy server. For more information, see <u>Team Foundation Server Permissions</u>.

# **Using TFSConfig Proxy**

TFSConfig Proxy /add|delete|change [/Collection:TeamProjectCollectionURL /account:AccountName] /Server:TeamFoundationServerURL [/Continue]

Argument	Description
TeamProjectCollectionURL	Used with /Collection. Specifies the URL of the team project collection that is hosted on the Team Foundation Service, in AccountName.DomainName/CollectionName format.
AccountName	Used with /Collection. Specifies the name of the account that is used as the service account for Team Foundation Service. If the account name has spaces, the name must be enclosed within quotation marks (""). All special characters in account names must be specified in accordance with command-line syntax.
ServerURL	Used with /Server. Specifies the URL of a TFS deployment, in ServerURL:Port/tfs format.
Option Description	
Adds the specified server or collection to the proxy list in the Proxy.config file. You can run /add multiple times to include more collections	

/add	or servers. When using /add with a collection hosted in Team Foundation Service, you will be prompted for your credentials in Team Foundation Service.
/change	Changes the credentials stored as the service account for Team Foundation Service. The /change option is only used for Team Foundation Service; it should not be used for local deployments of TFS.
/delete	Removes the specified server or collection from the proxy list in the Proxy.config file.
/account	Specifies the account used as the service account for the proxy in Team Foundation Service. This option is only used for Team Foundation Service in conjunction with the /change option.
	The default service account used for Team Foundation Service is "Account Service."
/continue	Continues the execution of the command even if the

verification process produces warnings.

# **Remarks**

You use the **Proxy** command to update the existing configuration of Team Foundation Server Proxy. You cannot use the Proxy command for initial installation and configuration of the proxy.

# Example

The following example shows how to add a TFS deployment named FABRIKAM to the proxy list.

#### ⊠Note

Examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

TFSConfig Proxy /add /Server:http://www.fabrikam.com:8080/tfs

The following example shows how to add a team project collection hosted on Team Foundation Service to the proxy list. The collection is named PhoneSaver and the account name used for Team Foundation Service is HelenaPetersen.fabrikam.com. Because the /account option is not specified, the default service account will be used.

### Copy Code

TFSConfig Proxy /add /Collection:https://HelenaPetersen.tfs.visuals1

The following example shows how to change the service account used by the proxy for the team project collection hosted on Team Foundation Service. The collection is named PhoneSaver, the account name used for Team Foundation Service is HelenaPetersen.fabrikam.com, and the service account used by the proxy is being changed to "My Proxy Service Account". Because the account name contains spaces, quotation marks are used to enclose the name.

#### 

TFSConfig Proxy /change /Collection:https://HelenaPetersen.tfs.visua

## See Also

## Concepts

How to: Install Team Foundation Proxy and Set Up a Remote Site

### **Other Resources**

Using Team Foundation Server Command-Line Tools

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
RebuildWarehouse Command
Example See Also Send Feedback

You can use the **RebuildWarehouse** command to rebuild the SQL Server Reporting Services and SQL Server Analysis Services databases that Visual Studio Team Foundation Server (TFS) uses.

To start the rebuild of these databases interactively, you can use the Reporting node in the administration console for Team Foundation. For more information, see <u>Open the Team Foundation Administration Console</u>.

**Required Permissions** 

To use the **RebuildWarehouse** command, you must be a member of the following groups:

- the Team Foundation Administrators security group and the Administrators security group on the server or servers that are running the administration console for Team Foundation
- the sysadmin group on the server or servers that are running the instance of SQL Server that hosts the databases for TFS

For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

TFSConfig RebuildWarehouse /analysisServices | /all [/ReportingDatas

## **Parameters**

Placeholder	Description	
Password	Specifies the password that the data sources account for SQL Server Reporting Services uses. This account is referred to in the documentation as TFSReports.	
	Option	Description
/analysisSei	vices	Required if <b>/all</b> is not used. Specifies that only the database for Analysis Services will be rebuilt. If no database exists for Analysis Services, you must also use the <b>/reportingDataSourcePassword</b> option.
/all		Required if <b>/analysisServices</b> is not used. Specifies that all reporting and analysis databases that TFS uses will be rebuilt.
		Required if the TFS_Analysis database does not exist. Specifies the password of the account that

/reportingDataSourcePassword:Password Password:Password Reporting Services (TFSReports). For more information, see Service Accounts and Dependencies in Team Foundation Server.

# **Remarks**

You might use this command when moving or splitting a team project collection, restoring data, or otherwise changing the configuration of your deployment.

# Example

The following example shows how to rebuild the Analysis Services database for a deployment of TFS.

#### ĭNote

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

TFSConfig RebuildWarehouse /analysisServices

TFSConfig - Team Foundation Server Configuration Tool Copyright © Microsoft Corporation. All rights reserved. The Analysis Services database was successfully rebuilt.

## See Also

### Tasks

Add a Report Server to Your Deployment

## Concepts

<u>Team Foundation Server Architecture</u> Understanding SQL Server and SQL Server Reporting Services <u>The Team Foundation Administration Console</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Recover Command
Send Feedback

The Recover command has been deprecated. Do not use it.

If you need to recover your deployment after a disaster, consult the following resources, or contact <u>Microsoft Support</u>:

- <u>Restoring a Single-Server Deployment to New Hardware [Tutorial]</u>
- Restore Data to the Same Location
- Organizing Your Server with Team Project Collections

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
RegisterDB Command
Example See Also Send Feedback

Use **RegisterDB** to update name of the server that hosts the configuration database in Visual Studio Team Foundation Server (TFS). You might use this command when restoring the configuration database to new hardware or when changing the domain of a deployment.

### **▲Important**

For the RegisterDB command to succeed, the following application pools and programs must be running:

- Team Foundation Server Application Pool (application pool)
- ReportServer (application pool)
- SQL Server Reporting Services (program)

In addition, the following components must not be open or running:

#### **Required Permissions**

To use the **RegisterDB** command, you must be a member of the Team Foundation Administrators group on the application-tier server for Team Foundation and a member of the sysadmin group in SQL Server on the data-tier server for Team Foundation. For more information, see <u>Team Foundation Server</u> <u>Permissions</u>.

### ⊠Note

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

TFSConfig RegisterDB /SQLInstance:ServerName /databaseName:DatabaseN

## **Parameters**

Placeholder	Descript	ion
ServerName	erverName Specifies the name of the server that hosts the configuration database for TFS and the name of the instance if you want to use an instance other than the default instance. If you specify an instance, you must use the following format:	
	ServerName\Inst	anceName
DatabaseName Specifies the name of the configuration database that you want to register with TFS.		
Arg	ument	Description
/SQLInstance:	ServerName	Required. Specifies the name of the server that is running SQL Server and the name of the instance if you want to use an instance other than the default instance. If you specify an instance, you

	must use the following format:
	ServerName\InstanceName
/ <b>databaseName:</b> DatabaseName	Required. Specifies the name of the configuration database. By default, this is Tfs_Configuration.
/usesqlalwayson	Optional. Specifies that the databases are part of an AlwaysOn Availability Group in SQL Server. If configured, this option sets MultiSubnetFailover in the connection string.
	For more information, see <u>AlwaysOn Availability</u> <u>Groups (SQL Server)</u> .

## **Remarks**

## **▲Important**

Back up the databases for TFS before you use this command.

You must provide the exact name or address of the configuration database for this command to operate correctly. If you must change the server on which this database is stored, you must ensure that TFS points to the new location.

# Example

The following example redirects TFS to a configuration database that is located on the server ContosoMain in the SQL Server instance TeamDatabases.

#### ĭNote

The examples are for illustration only and are fictitious. No real association is intended or inferred.

## Copy Code

TFSConfig RegisterDB /SQLInstance:ContosoMain\TeamDatabases /databases /datab

## See Also

### **Other Resources**

Managing Server Configuration with TFSConfig Using Team Foundation Server Command-Line Tools Team Foundation Server Services and Service Accounts

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
RemapDBs Command
Example See Also Send Feedback

The **RemapDBs** command redirects Visual Studio Team Foundation Server (TFS) to its databases when they are stored on more than one server and you are restoring, moving, or otherwise changing the configuration of your deployment. For example, you must redirect TFS to any databases for team project collections if they are hosted on a separate server or servers from the configuration database. You must also redirect TFS to the server or servers that are running SQL Server Analysis Services or SQL Server Reporting Services if those databases are hosted on a separate server or instance from the configuration database.

#### **Required Permissions**

To use the **RemapDBs** command, you must be a member of the Team Foundation Administrators security group and a member of the sysadmin security group for any SQL Server databases that Team Foundation Server uses. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

TFSConfig RemapDBs /DatabaseName:ServerName;DatabaseName /SQLInstanc

## **Parameters**

Argument	Description
ServerName	Specifies the name of the server that you want to map as a server that hosts one or more databases for TFS. If an instance other than the default instance hosts a database, you must also specify the name of the instance in the following format:

Specifies the name of the DatabaseName database that you are mapping to TFS.

## Option

## Description

Specifies the name of the server that hosts the database that you want to map for TFS, in addition to the name of the database itself.

/DatabaseName

/ <b>SQLInstances:</b> ServerName1,ServerName	Specifies the name of the server that is running SQL Server, in addition to the name of the instance if you want to use an instance other than the default instance.
	If you are specifying more than one server, you must use a comma to separate multiple pairs of server and instance names.
/ <b>AnalysisInstance:</b> ServerName	Optional. Specifies the name of the server and instance that hosts SQL Server Analysis Services. Use this option to specify the server and instance that hosts the Analysis Services database.
/ <b>AnalysisDatabaseName:</b> DatabaseName	Optional. Specifies the name of the Analysis Services database that you want to use with TFS if you have

## more than one such database on the server that you specified with the /AnalysisInstance option. **Optional**. Displays the actions that you /preview must take to update the configuration. **Optional.** Specifies that the **RemapDB** command should continue even if an error occurs during the attempt to locate one or more databases. If you use the /continue option, any /continue collections whose databases are not found on the server or servers that you specify will be reconfigured to use the server and instance that hosts the configuration database.

Optional. Specifies that the databases are part of an

## /usesqlalwayson

AlwaysOn Availability Group in SQL Server. If configured, this option sets MultiSubnetFailover in the connection string.

For more information, see <u>AlwaysOn</u> <u>Availability Groups</u> (SQL Server).

# **Remarks**

You use the **RemapDBs** command to reconfigure TFS to use different servers and instances of SQL Server from the servers and instances in the original installation.

# Example

The following example shows how to redirect TFS to its configuration database TFS\_Configuration. This database is hosted on ContosoMain on the named instance TeamDatabases. Its project collection databases are stored on both ContosoMain\TeamDatabases and the default instance on Contoso2.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

Сору

TFSConfig RemapDBs /DatabaseName:ContosoMain\TeamDatabases;TFS\_Confi

## See Also

## **Other Resources**

Managing Server Configuration with TFSConfig

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Repair Command
See Also Send Feedback

The Repair command has been deprecated. Do not use it without express guidance from Microsoft Support.

If you need to repair your stored procedures after a failed database patch, contact <u>Microsoft Support</u>.

## See Also

### Concepts

Team Foundation Server Architecture Backing up and Restoring Your Deployment The Team Foundation Administration Console

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Settings Command
Example See Also Send Feedback

You can use the **Settings** command to automate changes to the uniform resource locator (URL) that is used by the notification interface and for the server address for Visual Studio Team Foundation Server (TFS). By default, the notification interface URL and the server URL match in TFS, but you can configure separate URLs. For example, you might want to use a different URL for the automatic emails that TFS generates. You must run this tool from the application tier to update all servers so that they use the new URLs.

To change these URLs interactively or to just view the current settings, you can use the administration console for Team Foundation. For more information, see <u>Open the Team Foundation Administration Console</u>.

#### **Required Permissions**

To use the **Settings** command, you must be a member of the Team Foundation Administrators security group and the Administrators group on the application-tier server for Team Foundation. For more information, see <u>Team Foundation</u> <u>Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

TFSConfig Settings [/ServerURL:URL] [/NotificationURL:URL]

## **Parameters**

Placeholder	Des	cription
URL	Use to set the uniform resource locator (URL) with the appropriate option.	
Opt	ion	Description
/ServerURL	:URL	Specifies the URL of the application-tier server for Team Foundation. This value is stored in the configuration database for Team Foundation.
/Notificatio	<b>URL:</b> URL	Specifies the URL to use in the text of e- mail alerts, if that URL differs from the URL of the application- tier server for

Team Foundation. This value is stored in the configuration database.

# **Remarks**

The **Settings** command changes connection information for server-to-server communication in a deployment of TFS. The URL that is specified in **/ServerURL** must be available to all servers within the deployment.

# Example

The following example changes the value of NotificationURL to http://contoso.example.com/tfs and the value of ServerURL to http://contoso.example.com:8080/tfs.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

## Copy Code

TFSConfig Settings /NotificationURL:http://contoso.example.com/tfs ,
## **Other Resources**

<u>Using Team Foundation Server Command-Line Tools</u> <u>Team Foundation Server Services and Service Accounts</u>

Visual Basic C#
Visual C++
F#
HLSL
JScript
Updates Command
Example See Also Send Feedback

You can use the **Updates** command to install any software updates that are missing from the databases for Visual Studio Team Foundation Server (TFS). In most cases, software update mismatches will not occur, because updates are usually applied to the entire deployment. To operate correctly, the application tier and the databases must have the same software updates applied. If a software update mismatch occurs where the databases are missing one or more updates that have been applied to the application tier, you can use the **Updates** command to attempt to automatically apply the missing updates to the databases.

#### ⊠Note

The Updates command applies updates only to the databases. If a software update mismatch occurs where the databases have a software update that has not been applied to the application tier, you must manually apply that update to the application tier. For more information about how to determine whether a software update mismatch might have occurred, see <u>Diagnose Command</u>.

#### **Required Permissions**

To use the **Updates** command, you must be a member of the Team Foundation Administrators group on the application-tier server for Team Foundation and a member of the sysadmin group in SQL Server on the data-tier server for Team Foundation. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

TFSConfig Updates /reapply

## Option Description

Specifies that any software updates that have been applied to the application tier /**reapply** for Team Foundation will be applied to the databases for TFS, if the software updates are not already present.

# **Remarks**

You use the **Updates** command to attempt to automatically apply to the TFS databases any software updates that have already been applied to the application tier. When you run this command, messages will inform you about the status of the update attempts. Update jobs will be queued for any databases that are missing the updates. These databases might include the configuration database, the warehouse database, and all databases for team project collections. If one or more jobs cannot complete, error messages will specify which databases could not be updated and provide more information, if any is available.

# Example

The following example shows how to run the **Updates** command to apply to the databases for Team Foundation Server any software updates that have already been applied to the application tier.

## Copy Code

TFSConfig Updates /reapply

### Reference

**Diagnose Command** 

## Concepts

Team Foundation Server Architecture

### **Other Resources**

Upgrading Team Foundation Server

Visual Basic C#
Visual C++
F#
HLSL
JScript
Configuring Lab Management with TFSLabConfig
See Also Send Feedback

Team Foundation Server includes a command-line tool to help you configure and manage the lab service provide by Visual Studio Lab Management.

The **TFSLabConfig** command-line tool is located in Drive:\Program Files\Microsoft Team Foundation Server 2010\Tools on Team Foundation Server application tier. It is also located in Drive:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE on the client machine where Microsoft Test Manager is installed.

**Required Permissions** 

To use TFSLabConfig, you must have the appropriate permissions for the operation that you want to perform. The required permissions are described for each command in the command reference topic.

# **Commands**

This tool provides commands to perform the following tasks.

Task	• Command
Manage host groups in a team project. Add or remove a host group in an individual team project.	<ul> <li>FSLabConfig CreateTeamProjectHostGroup Command</li> <li>TFSLabConfig DeleteTeamProjectHostGroup Command</li> <li>TFSLabConfig ListTeamProjectHostGroups Command</li> <li>TFSLabConfig ListTeamProjectCollectionHostGroups Command</li> <li>TFSLabConfig TPCHostGroup Command</li> <li>TFSLabConfig TPCLibraryShare Command</li> </ul>
Manage library shares in a team project. Add or remove a library share in an individual team project in the	<ul> <li>TFSLabConfig CreateTeamProjectLibraryShare Command</li> <li>TFSLabConfig DeleteTeamProjectLibraryShare Command</li> <li>TFSLabConfig ListTeamProjectLibraryShares Command</li> <li>TFSLabConfig ListTeamProjectCollectionLibraryShares Command</li> </ul>

collection.	<u>TFSLabConfig TPCLibraryShare Command</u>
Manage permissions for users on specified Lab Management objects. Add or remove permissions for users and user groups to an object in a team project or team project collection.	• TFSLabConfig Permissions Command
Manage the service account of a team project collection. Add or modify the user account credential that is used by test agents and build agents to communicate with test and build	<ul> <li>TFSLabConfig SetServiceAccount Command</li> <li>TFSLabConfig GetServiceAccount Command</li> <li>TFSLabConfig ClearServiceAccount Command</li> <li>TFSLabConfig UpdateServiceAccountOnDeployedEnvironments Command</li> </ul>

controllers.

### Concepts

<u>Configuring Your Server Using the Team Foundation Administration Console</u> <u>Configuring and Administering Lab Management</u> TFSConfig Lab Commands

#### **Other Resources**

**Using Team Foundation Server Command-Line Tools** 

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig CreateTeamProjectHostGroup Command Example See Also Send Feedback

Use the **CreateTeamProjectHostGroup** command to assign a host group from a team project collection to an individual team project in the collection. Host groups specify one or more physical machines that are the deployment targets for virtual environments in Visual Studio Lab Management. Host groups are created in System Center Virtual Machine Manager (SCVMM) and assigned to a project collection by Visual Studio Lab Management. Use separate **CreateTeamProjectHostGroup** commands to assign multiple host groups to a team project.

#### ⊠Note

You can automatically assign a host group to all the projects in a team project collection when you assign the host group to the team project collection. See <u>TFSConfig Lab /HostGroup Commands</u> and <u>How to: Change the Host Groups</u> for Your Team Project Collections.

#### **Required Permissions**

To use the **CreateTeamProjectHostGroup** command, you must have the **Manage Lab Locations** permission at the Team Project Collection Host Group level. By default, the members of the Team Foundation Server Administrators and Project Collection Administrators groups have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

```
TFSLabConfig CreateTeamProjectHostGroup
    /Collection:collectionUrl
    /TeamProject:{* |teamProjectName}
    /TeamProjectCollectionHostGroup:(* |teamProjectCollectionHost(
```

/Name:teamProjectHostGroupName [/Description:teamProjectHostGroupDescription] [/NoPrompt]

Option	Description
<b>Collection</b> :collectionUrl	Required. The URL of the team on the application tier of Team F that contains the team project. F /collection:http://abc:8080/TFS/.
<b>TeamProject:</b> {*  teamProjectName>}	Required. The name of the team quotation marks if there are spac Use an asterisk (* ) to assign the group to all team projects in the
<b>TeamProjectCollectionHostGroup:</b> {*  teamProjectCollectionHostGroupName}	Required. The name of the host project collection. Use quotation are spaces in the name. Use an a assign all host groups in the coll specified team project. Use quot there are spaces in the name.
Name:teamProjectHostGroupName	Required. The name to assign to the team project.
<b>Description:</b> teamProjectHostGroupDescription	Optional. A description of the te group.
NoPrompt	Optional. Do not prompt the use confirmation.

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

In the first example, all the host groups in the team project collection are assigned to each team project in the collection. In the second example, one host group in the team project collection is assigned to a specific team project.

#### Copy Code

### Reference

TFSLabConfig ListTeamProjectCollectionHostGroups Command TFSLabConfig ListTeamProjectHostGroups Command TFSLabConfig DeleteTeamProjectHostGroup Command

### Concepts

TFSLabConfig Command Reference <u>Configuring Your Server Using the Team Foundation Administration Console</u> TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 TFSLabConfig CreateTeamProjectLibraryShare Command
 Example See Also Send Feedback

Use the **CreateTeamProjectLibraryShare** command to assign a library share from a team project collection to an individual team project in the collection. A library share provides access to file-based resources for virtual environments such as ISO images and virtual hard disks. Library shares are created in System Center Virtual Machine Manager (SCVMM) and assigned to project collection by Visual Studio Lab Management. Use separate

**CreateTeamProjectLibraryShare** commands to assign multiple library shares to a team project.

#### ⊠Note

You can automatically assign a library share to all projects in a team project collection by using the <u>TFSConfig Lab /LibraryShare Commands</u> and <u>How to:</u> <u>Change the Library Share for Your Team Project Collections</u>.

#### **Required Permissions**

To use the **CreateTeamProjectLibraryShare** command, you must have **Manage Lab Locations** permission at the Team Project Collection Library Share level. By default, members of the Team Foundation Server Administrators and Project Collection Administrators groups have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

```
TFSLabConfig CreateTeamProjectLibraryShare
Collection:collectionUrl
    /TeamProject:{* |teamProjectName}
    /TeamProjectCollectionLibraryShare:{* |teamProjectCollectionLi
    /Name:teamProjectLibraryShareName
```

[/Description:teamProjectLibraryShareDescription]
[/NoPrompt]

Option	Description
<b>Collection:</b> collectionUrl	Required. The URL of the team proje on the application tier of Team Found Server that contains the team project. example, /collection:http://abc:8080/TFS/Defa
<b>TeamProject:</b> {*  teamProjectName}	Required. The name of the team proj- quotation marks if there are spaces in Use an asterisk (*) to assign all librar the collection to the specified team p
<b>TeamProjectCollectionLibraryShare:</b> {*  teamProjectCollectionLibraryShareName}	Required. The name of the library shote team project collection. Use an asteri create a team project library share for shares in the team project collection.
Name:teamProjectLibraryShareName	Required. The name to assign to the ] in the team project.
<b>Description:</b> teamProjectLibraryShareDescription	Optional. A description of the team p share.
NoPrompt	Optional. Do not prompt the user for confirmation.

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

In the first example, all the library shares in the team project collection are assigned to each of the team projects in the collection. In the second example, one library share in the team project collection is assigned to a specific team project.

#### Copy Code

```
REM First example
TFSLabConfig CreateTeamProjectLibraryShare
    /collection:http://abc:8080/TFS/Collection0
    /TeamProject:*
    /TeamProjectCollectionLibraryShare:*
```

```
REM Second example
TFSLabConfig CreateTeamProjectLibraryShare
```

```
/collection:http://abc:8080/TFS/Collection0
/TeamProject:Project1
/TeamProjectCollectionLibraryShare:tpcls1
/name:ls1
```

### Reference

TFSLabConfig ListTeamProjectCollectionLibraryShares Command TFSLabConfig ListTeamProjectLibraryShares Command TFSLabConfig DeleteTeamProjectLibraryShare Command

### Concepts

TFSConfig Lab Command Reference <u>Configuring Your Server Using the Team Foundation Administration Console</u> TFSLabConfig Command Reference <u>Configuring and Administering Lab Management</u>

 Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 TFSLabConfig DeleteTeamProjectHostGroup Command Example See Also Send Feedback

Use the **DeleteTeamProjectHostGroup** command to remove the assignment of a host group from an individual team project. Host groups specify one or more physical machines that are the deployment targets for virtual environments in Visual Studio Lab Management.

Host groups are created in System Center Virtual Machine Manager (SCVMM). In Lab Management, host groups are assigned to one or more team project collections and then to one or more team projects in the collections. The **DeleteTeamProjectHostGroup** command does not remove the assignment of the host group to the team project collection.

**Required Permissions** 

To use the **DeleteTeamProjectHostGroup** command, you must have Delete Lab Locations permission for the Team Project host group. By default, Team Foundation Server Administrators, Team Project Collection Administrators and Team Project Administrators have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

TFSLabConfig DeleteTeamProjectHostGroup
 /Collection:collectionUrl
 /TeamProject:{\* |teamProjectName>}
 /Name:{\* |teamProjectHostGroupName}
 [/NoPrompt]

Option	Description
<b>Collection</b> :collectionUrl	Required. The URL of the team project collection on the application tier of Team Foundation Server that contains the team project. For example, /collection:http://abc:8080/TFS/DefaultCollectior
<b>TeamProject:</b> {*  teamProjectName}	Required. The name of the team project from which you want to remove the host group. Use quotation marks if there are spaces in the name. Use an asterisk (*) to specify all team projects in the team project collection.
<b>Name:</b> {*  teamProjectHostGroupName}	Required. The name of the host group to delete from a team project. Use quotation marks if there are spaces in the name. Use an asterisk (*) to specify all host groups in the team project.
NoPrompt	Optional. Do not prompt the user for confirmation.

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

In the first example, all the host groups assigned to each of the team project in the team project collection are removed. In the second example, one host group is removed from a specific team project.

#### Copy Code

```
REM First example
TFSLabConfig DeleteTeamProjectLibraryShare
    /collection:http://abc:8080/TFS/DefaultCollection
    /teamProject:*
    /name:*
REM Second example
TFSLabConfig DeleteTeamProjectLibraryShare
    /collection:http://abc:8080/TFS/DefaultCollection
    /teamProject:Project1
    /name:hg1
```

### Reference

TFSLabConfig ListTeamProjectCollectionHostGroups Command TFSLabConfig ListTeamProjectHostGroups Command TFSLabConfig CreateTeamProjectHostGroup Command

### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 TFSLabConfig DeleteTeamProjectLibraryShare Command
 Example See Also Send Feedback

Use the **DeleteTeamProjectLibraryShare** command to remove the assignment of a library share from an individual team project. A library share provides access to file-based resources for virtual environments such as ISO images and virtual hard disks., Library shares are created by System Center Virtual Machine Manager (SCVMM). In Visual Studio Lab Management, library shares are assigned to one or more team project collections and then to one or more team projects in the collections. The **DeleteTeamProjectLibraryShare** command does not remove the assignment of the library share to the team project collection.

**Required Permissions** 

To use the **DeleteTeamProjectLibraryShare** command, you must have **Delete Lab Locations** permission for that Team Project Library Share. By default, members of the Team Foundation Server Administrators, Project Collection Administrators and Team Project Administrators groups have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

```
TFSLabConfig DeleteTeamProjectLibraryShare
Collection:collectionUrl
   /TeamProject:{* |teamProjectName}
   /Name:{* |teamProjectLibraryShareName}
   [/NoPrompt]
```

Argument	Description
Collection:collectionUrl	Required. The URL of the team project collectic that contains the team project. For example, /collection:http://abc:8080/TFS/DefaultCollectic
<b>TeamProject:</b> {*  teamProjectName}	Required. The name of the team project that contains the library share that you want to delete Use quotation marks if there are spaces in the name. Use an asterisk (*) to specify all team projects in the team project collection.
<b>Name:</b> (*  teamProjectLibraryShareName)	Required. The name of the library share to delet from a team project. Use quotation marks if the are spaces in the name. Use an asterisk (*) to specify all library shares of the team project n.
NoPrompt	Optional. Do not prompt the user for confirmation.

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

In the first example, all the library shares assigned to each of the team project in the team project collection are removed. In the second example, one library share is removed from a specific team project.

#### Copy Code

```
REM First example
TFSLabConfig DeleteTeamProjectLibraryShare
    /collection:http://abc:8080/TFS/DefaultCollection
    /teamProject:*
    /name:*
REM Second example
TFSLabConfig DeleteTeamProjectLibraryShare
    /collection:http://abc:8080/TFS/DefaultCollection
    /teamProject:Project1
    /name:ls1
```

## Reference

TFSLabConfig ListTeamProjectCollectionLibraryShares Command TFSLabConfig ListTeamProjectLibraryShares Command TFSLabConfig CreateTeamProjectLibraryShare Command

### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 TFSLabConfig ListTeamProjectCollectionHostGroups Command
 Example See Also Send Feedback

Use the **ListTeamProjectCollectionHostGroups** command to list the System Center Virtual Machine Manager (SCVMM) host groups that are assigned to a team project collection and that you have read access to. These collection host groups can be added to a team project by using the **CreateTeamProjectHostGroup** command. For more information, see <u>TFSLabConfig CreateTeamProjectHostGroup Command</u>

**Required Permissions** 

To use the **ListTeamProjectCollectionHostGroups** command, you must have **View Lab Resources** permission at the Team Project collection level. By default, members of the TFS Administrators and Team Project Collection Administrators groups have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

Option

## Description

**Collection:**collectionUrl Required. The URL of the team project collection on the application tier of Team Foundation Server. For example, /collection:http://abc:8080/TFS/DefaultCollection.

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

#### Copy Code

TFSLabConfig ListTeamProjectCollectionHostGroups
 /collection:http://abc:8080/TFS/Collection0

### Reference

TFSLabConfig ListTeamProjectHostGroups Command TFSLabConfig CreateTeamProjectHostGroup Command TFSLabConfig DeleteTeamProjectHostGroup Command

#### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Visual Basic C#
 Visual C++
 F#
 HLSL
 JScript
 TFSLabConfig ListTeamProjectCollectionLibraryShares Command
 Example See Also Send Feedback

Use the **ListTeamProjectCollectionLibraryShares** command to list the library shares that have been assigned to a team project collection and that you have read access to. Library shares are created in System Center Virtual Machine Manager (SCVMM). and are assigned to a team project collection by using the <u>TFSConfig Lab /LibraryShare Commands</u> Team project collection library shares can be added to a team project by using the **CreateTeamProjectLibraryShare** command. For more information, see <u>TFSLabConfig</u> <u>CreateTeamProjectLibraryShare Command</u>.

**Required Permissions** 

To use the **ListTeamProjectCollectionLibraryShares** command, you must have **View Lab Resources** permission at the Team Project collection level. By default, members of the TFS Administrators and Team Project Collection Administrators groups have this permission. For more information, see <u>Team</u> <u>Foundation Server Permissions</u>.

TFSLabConfig ListTeamProjectCollectionLibraryShares
Collection:collectionUrl

#### Argument

## Description

**Collection:**collectionUrl Required. The URL of the team project collection on the application tier of Team Foundation Server. For example, /collection:http://abc:8080/TFS/DefaultCollection.
# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

#### Copy Code

TFSLabConfig ListTeamProjectCollectionLibraryShares
 /collection:http://abc:8080/TFS/Collection0

#### Reference

TFSLabConfig ListTeamProjectLibraryShares Command TFSLabConfig CreateTeamProjectLibraryShare Command TFSLabConfig DeleteTeamProjectLibraryShare Command

#### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Send feedback on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig ListTeamProjectHostGroups Command
Example See Also Send Feedback

Use the **ListTeamProjectHostGroups** command to list the host groups that are assigned to a team project and that you have read access to.

**Required Permissions** 

To use the ListTeamProjectHostGroups command, you must have **View Lab Resources** permissions at the Team Project level. By default, members of the Team Foundation Server Administrators, Team Project Collection Administrators, Team Project Administrators, Team Project Contributors, and Team Project Readers groups have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

TFSLabConfig ListTeamProjectHostGroups Collection:collectionUrl /TeamProject:teamProjectName

### **Parameters**

Argument	Description
<b>Collection:</b> collectionUrl	Required. The URL of the team project collectic on the application tier of Team Foundation Server that contains the team project. For example, /collection:http://abc:8080/TFS/DefaultCollectic
TeamProject:teamProjectName	Required. The name of the team project. Use quotation marks if there are spaces in the name

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

#### Copy Code

TFSLabConfig ListTeamProjectHostGroups
 /collection:http://abc:8080/TFS/DefaultCollection
 /teamProject:Project1

#### Reference

TFSLabConfig ListTeamProjectCollectionHostGroups Command TFSLabConfig CreateTeamProjectHostGroup Command TFSLabConfig DeleteTeamProjectHostGroup Command

#### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig ListTeamProjectLibraryShares Command See Also Send Feedback

Use the **ListTeamProjectLibraryShares** command to list all library shares that are assigned to a team project and that you have access to.

**Required Permissions** 

To use the ListTeamProjectLibraryShares command, you must have the **View Lab Resources** permission at Team Project level. By default, members of the TFS Administrators, Team Project Collection Administrators, Team Project Administrators, Team Project Contributors, and Team Project Readers groups have this permission. For more information, see <u>Team Foundation Server</u> <u>Permissions</u>.

```
TFSLabConfig ListTeamProjectLibraryShares
Collection:collectionUrl
     /TeamProject:teamProjectName
```

### **Parameters**

Option	Description
<b>Collection:</b> collectionUrl	Required. The URL of the team project collectic on the application tier of Team Foundation Server that contains the team project. For example, /collection:http://abc:8080/TFS/DefaultCollectic
TeamProject:teamProjectName	Required. The name of the team project. Use quotation marks if there are spaces in the name.

#### Reference

TFSLabConfig ListTeamProjectCollectionLibraryShares Command TFSLabConfig DeleteTeamProjectLibraryShare Command TFSLabConfig CreateTeamProjectLibraryShare Command

#### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig Permissions Command
See Also Send Feedback

Use the **TFSLabConfig Permissions** command to set and get permissions set for a specified user or for multiple users on a specified object in Visual Studio Lab Management. For more information about individual permissions, see the Lab Management Permissions section of <u>Team Foundation Server Permissions</u>.

**Required Permissions** 

To query permissions on an object, you must have Read permissions for the object. To change permissions on an object the **Permissions** command, you must have the **Manage Permissions** permission. By default, the creator of the object has this permission. For more information, see <u>Team Foundation Server</u> <u>Permissions</u>.

```
TFSLabConfig Permissions
    /Collection:collectionUrl
    [objectSpec]
    {[/User:userName1[,userName2][,...]]}
    [/Group:groupName1[,groupName2][,...]}}
    [/Allow:{* |perm1[,perm2][,...]}]
    [/Deny:{* |perm1[,perm2][,...]}]
    [/Remove:{* |perm1[,perm2][,...]}]
    [/Inherit:Yes|No]
```

### **Parameters**

Option	Description
<b>Collection:</b> collectionUrl	Required. The URL of the team project colle on the application tier of Team Foundation 5 For example, /collection:http://abc:8080/TFS/DefaultColl
objectSpec	Optional. Specifies the target object such as team project or library share to which the permissions are applied. For information ab- how to specify objects, see <u>objectSpec</u> below
User:userName1[,userName2][,]	Optional. Specifies one or more users to wh permissions are applied. Use commas to sep multiple user names.
<b>Group:</b> groupName1[,groupName2] [,]	Optional. Specifies one or more groups to w the permissions are applied. Use commas to separate multiple group names.
<b>Allow:</b> {*  perm1[,perm2][,]}	Optional. Enables the specified permissions the specified users or groups. Use an asterist to specify all permissions. To specify an individual permission, use the identifiers in <b>Name at command line</b> column of the table the Lab Management Permissions section of <u>Foundation Server Permissions</u> . Use comma separate multiple permissions.

<b>Deny:</b> {*  perm1[,perm2][,]}	Optional. Denies the specified permissions f specified users or groups. Use an asterisk (* specify all permissions. To specify an indivi permission, use the identifiers in the <b>Name</b> <b>command line</b> column of the table in the La Management Permissions section of <u>Team</u> <u>Foundation Server Permissions</u> . Use comma separate multiple permissions.
<b>Remove:</b> {*  perm1[,perm2][,]}	Optional. Unsets the specified permissions t were previously granted or denied to the use group. To specify an individual permission, the identifiers in the <b>Name at command lin</b> column of the table in the Lab Management Permissions section of <u>Team Foundation Se</u> <u>Permissions</u> . Use commas to separate multij permissions.
Inherit:Yes No	Optional. If you specify <b>Yes</b> , all permissions associated with a parent ACL are inherited bitem. Cannot be combined with the / <b>remove</b> / <b>user</b> , or / <b>group</b> options.

# ∎objectSpec

You can specify the objects that you want to include in the **TFSLabConfig Permissions** command in two ways:

- Use one or more locations options to specify the object in the Lab Management hierarchy.
- Use the /**Url** option to specify the object as a Uri.

If the objectSpec parameter option is not specified, the permissions are applied to all objects in the team project collection.

### **Object type options**

The following table lists the valid combination of options that you can use to specify an object as the objectSpec parameter of a **TFSLabConfig permissions** command.

To set permissions on	Use these options
A specific host group in a team project collection	/ <b>TeamProjectCollectionHostGroup:</b> teamProjectCollectionHostGr
A specific library share in a team project collection	/TeamProjectCollectionLibraryShare:teamProjectCollectionLibrationL

All group hosts in a team project	/ <b>TeamProject:</b> projectName / <b>TeamProjectHostGroup:*</b>
A group host in a team project	/ <b>TeamProject:</b> projectName / <b>TeamProjectHostGroup:</b> teamProjectHostGroupName
A lab environment in a host group for a team project	/ <b>TeamProject:</b> projectName / <b>TeamProjectHostGroup:</b> teamProjectHostGroupName / <b>LabEnvironment:</b> labEnvironmentName
All library shares in a team project	/TeamProject: projectName /TeamProjectLibraryShare:*
A library share in a team project	/ <b>TeamProject:</b> projectName / <b>TeamProjectLibraryShare:</b> teamProjectLibraryShareName
A lab template in a library share of a team project	/ <b>TeamProject:</b> projectName / <b>TeamProjectLibraryShare:</b> teamProjectLibraryShareName / <b>LabTemplate:</b> labTemplateName
A lab environment in a library share of a	/ <b>TeamProject:</b> projectName / <b>TeamProjectLibraryShare:</b> teamProjectLibraryShareName / <b>LabEnvironment:</b> labEnvironmentName

### ∎Url

Use the following syntax to specify the objectSpec target object of a **TFSLabConfig permissions** command by using the **/Url** option:

/url:VSTFS:///LabManagement/objectType/objectId

The objectId is the unique numeric identifier of the object.

The following table lists the valid keywords for the objectType keyword:

Object Type	Description
TeamProjectCollectionHostGroup	A host group of a team project collection
TeamProjectCollectionLibraryShare	A library share of a team project collection
TeamProject	A team project
TeamProjectHostGroup	A host group of a team project

TeamProjectLibraryShare	A library share of a team project
LabTemplate	A virtual machine or template in a team project library share
LabEnvironment	An environment that is deployed on a team project host group or stored in a team project library share.

# **Permissions**

You can specify one or more of Lab Management permissions as the target of the /Allow, /Deny, or /Remove options. For a lists of available permissions, see the Lab Management Permissions section of the

Team Foundation Server Permissions topic.

- Use an asterisk (\*) to specify all lab permissions.
- Use commas to separate multiple permissions.

# **Remarks**

If neither the **/User** or **/Group** option is specified, the current permissions of the specified object is displayed.

If the objectSpec parameter option is not specified, the permissions are applied to all objects in the team project collection.

#### Concepts

TFSLabConfig Command Reference <u>Team Foundation Server Permissions</u> TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u> Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig SetServiceAccount Command
See Also Send Feedback

Sets the service account credentials of a team project collection to the specified user in Visual Studio Team Foundation Server. The service account is added to the following team project collection groups:

- **Build Service** for deployment activities in the build agent and access to drop locations.
- **Test Service** for authorization of communication between test agents and test controllers.

The service account is used by test agents and build agents in a team project collection to communicate with test and build controllers.

#### ⊠Note

The credentials of the service account are not secured in Team Foundation Server. The service account should be created with the minimum rights on the system. The **SetServiceAccount** command grants the specified user the permissions that are required to enable communication between agents and controllers.

#### **Required Permissions**

To use the **SetServiceAccount** command, you must have the **Manage Lab Locations** permission at the Team Project Collection Host Group level. By default, the members of the TFS Administrators and Project Collection Administrators groups have this permission. For more information, see <u>Team</u> <u>Foundation Server Permissions</u>. TFSLabConfig SetServiceAccount /Collection:collectionURL /User:userName

# Options

Option	Description
<b>Collection:</b> collectionUrl	Required. The URL of the team project collection on the application- tier of Team Foundation Server to associate with the service account.
<b>User:</b> userName	Required. The domain, if necessary, and user name of the service account.

## **Remarks**

The SetServiceAccount command replaces any existing service account credentials in the team project collection configuration. However, the credentials that are used in deployed environments are not replaced. Use the UpdateCredentialsOnDeployedEnvironments command to replace the credentials in existing environments.

### Description

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

This example adds a service account for a team project collection.

### Code

#### Copy Code

```
TFSLabConfig SetServiceAccount
    /collection:http://contoso:8080/TFS/DefaultCollection
    /user:contoso\user1
```

#### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig GetServiceAccount Command
Example See Also Send Feedback

Lists the current service account that is used by test agents and build agents to communicate with the test and build controllers in a team project collection.

**Required Permissions** 

To use the **GetServiceAccount** command, you must have the **Manage Lab Locations** permission at the Team Project Collection Host Group level. By default, the members of the TFS Administrators and Project Collection Administrators groups have this permission. For more information, see <u>Team</u> <u>Foundation Server Permissions</u>.

TFSLabConfig GetServiceAccount /collection:collectionUrl

# Options

Option

Description

The URL of the team project collection on **Collection:**collectionUrl the applicationtier of Team Foundation Server.

# **Remarks**

The service account of a team project collection is set by the TfsLabConfig SetServiceAccount command or in the User Account tab of the Create Team Project Collection dialog box.

# Example

For increased readability in this example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

This example lists the user account for a team project collection.

#### Copy Code

tfslabconfig /getserviceaccount
 /collection:http://contoso:8080/TFS/DefaultCollection

#### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig ClearServiceAccount Command
Example See Also Send Feedback

Removes the service account credentials that are used by the test and build agents of a team project collection to communicate with the test controllers and build controllers.

**Required Permissions** 

To use the **ClearServiceAccount** command, you must have the **Manage Lab Locations** permission at the Team Project Collection Host Group level. By default, the members of the Team Foundation Server Administrators and Project Collection Administrators groups have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

TFSLabConfig ClearServiceAccount /collection:collectionURL

# **Options**

Option

Description

The URL of the team project collection on **Collection:**collectionUrl the application tier of Team Foundation Server.

# **Remarks**

The service account of a team project collection is set by the TfsLabConfig SetServiceAccount command or in the User Account tab of the Create Team Project Collection dialog box.

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

This example clears the user accounts for a team project collection.

#### Copy Code

tfslabconfig clearserviceaccount
 /collection:http:// contosotfs:8080/TFS/DefaultCollection

#### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

- Nisual Basic □ C#
  □ Visual C++
  □ F#
  □ HLSL
- □ ILOL
- JScript

TFSLabConfig UpdateServiceAccountOnDeployedEnvironments Command <u>Example</u> <u>See Also</u> <u>Send Feedback</u>

Adds or changes the service account of all or some of the existing environments in a team project collection to the current service account of the team project collection. The service account is used by test agents and build agents in a team project collection to communicate with test and build controllers.

**Required Permissions** 

To use the **UpdateServiceAccountOnDeployedEnvironments** command, you must have the **Manage Lab Locations** permission at the Team Project Collection Host Group level. By default, the members of the Team Foundation Server Administrators and Project Collection Administrators groups have this permission. For more information, see <u>Team Foundation Server Permissions</u>.

```
TFSLabConfig UpdateServiceAccountOnDeployedEnvironments
    /Collection:collectionUrl,
    [/LabEnvironmentIds:id1,id2,...]
```

### **Parameters**

Option	Description
<b>Collection:</b> collectionUrl	Required. The URL of the team project collection on the application tier of Team Foundation Server to associate with the credentials. that contains the team project. If the LabEnvironmentIds option is not specified, all environments in the team project collection are updated.
LabEnvironmentIds:id1,id2,	One or more comma-separated identifiers of the team project collection environments that you want to update. You can view the environment identifiers in the Environment
Viewer of Lab Management and from the XML description of the virtual machine in SCVMM Admin Console.

# **Remarks**

The service account of a team project collection are set by the TfsLabConfig SetServiceAccount command or in the User Account tab of the Create Team Project Collection dialog box.

# Example

For increased readability in the example, command options are listed on separate lines. In a command prompt window, type all options for a command on the same line.

The first example updates the service account of all environments in a team project collection. In the second example, only the specified environments are updated.

### Copy Code

REM First example TFSLabConfig UpdateServiceAccountOnDeployedEnvironments /collection:http://contosotfs:8080/TFS/DefaultCollection

REM Second example

TFSLabConfig UpdateServiceAccountOnDeployedEnvironments
 /collection:http:// contosotfs:8080/TFS/DefaultCollection
 /labEnvironmentIds:5,8

## See Also

### Concepts

TFSLabConfig Command Reference TFSConfig Lab Command Reference <u>Configuring and Administering Lab Management</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig TPCHostGroup Command
Example See Also Send Feedback

### ⊠Note

These commands only work on SCVMM 2012 server, and are not supported on SCVMM 2008 R2 server.

Use the **TPCHostGroup** command to add or remove a host group from a team project collection, or to modify the Lab Management settings of the host group.

To run these commands, you must be a member of Team Project Collection Administrators group in Team Foundation Server for the collection you specify. In addition, you must be a member of Administrator or Delegated Administrator role in the SCVMM Server from which you are adding the host groups. Furthermore, you must ensure that no other team project collection in any Team Foundation Server is already using the same SCVMM host group.

```
TfsLabConfig TPCHostGroup
    /collection:collectionUrl
    [/add
        /scvmmhostgroup:scvmmHostGroupPath
        /name:name
        [/dnssuffix:dnsSuffixForNetworkIsolation]
        [/autoprovision:{True|False}]]
    [/delete
        /name:name
        [/noprompt]]
    [/edit
        /name:hostGroupName
        [/autoprovision:{True|False}]
        [/dnssuffix:dnSuffixForNetworkIsolation]
        [/noprompt]]
```

### [/list] [/listscvmmhostgroups]

### **Parameters**

Option	Description
<b>collection:</b> collectionUrl	Required. Specifies the URL of the team project collection on the application-tier of the Team Foundation Server.
add	Adds the specified host group to the team project collection. You must specify the /collection, /scvmmhostgroup, and name options with /add.
<b>scvmmhostgroup:</b> scvmmHostGroupPath	Required with / <b>add</b> . Specifies the fully qualified domain name (FDQN) of the VMM host group. The FQDN path can be found by using the VMM Admin Console.

<b>name:</b> hostGroupName	Required with / <b>add</b> . Specifies the name of the host group in the team project collection.
<b>dnsSuffix:</b> dnsSuffixForNetworkIsolation	Optional with /add or /edit. Specifies the DNS suffix that will be used to register the names of virtual machines on the isolated network for the virtual environments within this host group. To confirm that the suffix is configured correctly in the DNS hierarchy, contact your network administrator.
	Optional with / <b>add</b> or / <b>edit</b> . Specifies whether the host group is automatically assigned to each team project in the collection. By default, / <b>autoProvision</b> is

autoProvision:True False	set to <b>True</b> , and host group is automatically assigned to every team project in the collection. Note: The / <b>autoProvision</b> option affects existing team projects.
Delete	Removes the specified host group from the team project collection. You must specify the / <b>collection</b> and / <b>name</b> options.
noPrompt	Optional with / <b>add</b> , / <b>edit</b> , or / <b>delete</b> . Suppresses display progress and result data from the command window.
edit	Changes the Lab Management settings of the host group. You must specify the / <b>collection</b> and / <b>name</b> options.

list

Lists all host groups that are assigned to the specified team project collection.

listVmmHostGroups

Lists all host groups that are available in VMM.

# Example

In this example, a host group is added to a team project collection:

### Copy Code

TFSLabConfig TPCHostGroup
 /collection:http://abc:8080/TFS/Collection0
 /name:HostGroup1

### See Also

### Reference

TFSLabConfig TPCLibraryShare Command

### Concepts

Configuring Lab Management with TFSLabConfig Configuring and Administering Lab Management

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSLabConfig TPCLibraryShare Command
Example See Also Send Feedback

### ⊠Note

These commands only work on SCVMM 2012 server, and are not supported on SCVMM 2008 R2 server.

Use the **TPCLibraryShare** command to add or remove a library share from a team project collection, or to modify the lab management settings of the library share.

To run these commands, you must be a member of Team Project Collection Administrators group in Team Foundation Server for the collection you specify. In addition, you must be a member of Administrator or Delegated Administrator role in the SCVMM Server from which you are adding the host groups. Furthermore, you must ensure that no other team project collection in any Team Foundation Server is already using the same SCVMM library share.

```
TfsLabConfig TPCLibraryShare
    /collection:collectionUrl
    [/add
        /scvmmLibraryShare:vmmLibrarySharePath
        /name:libraryShareName
             [/autoprovision:{True|False}]]
    [/delete
             /name:libraryShareName
             [/noprompt]]
    [/edit
             /name:libraryShareName
             [/autoprovision:{True|False}]
             [/noprompt]]
        [/list]
```

[/listscvmmlibraryshares]

## **Parameters**

Option	Description
<b>collection:</b> collectionUrl	Required. Specifies the URL of the team project collection on the application-tier of the Team Foundation Server.
add	Adds the specified library share to the team project collection. You must specify the / <b>collection</b> , / <b>scvmmlibraryshare</b> , and / <b>name</b> options with / <b>add</b> .
<b>scvmmLibraryShare:</b> vmmLibrarySharePath	Required with <b>/add</b> . Specifies the fully qualified domain name (FQDN) of the VMM share. The FDQN path can be retrieved by using the VMM Admin Console.
	Required with /add,

#### name:libraryShareName

#### autoProvision:True|False

#### Delete

edit, and delete. Specifies the name of the library share in the team project collection.

Optional with /add or /edit. Specifies whether the library share is automatically assigned to each team project in the collection. By default, /autoProvision is set to True, and library share is automatically assigned to every team project in the collection. The /autoProvision option only affects existing team projects.

Removes the specified library share from the team project collection. You must specify the /collection and /name options.

Optional with /**add**, /**edit**, or /**delete**. Suppresses display

noPrompt	progress and result data from the command window.
edit	Changes Lab Management settings of the library share. You must specify the / <b>collection</b> and / <b>name</b> options.
list	Lists all library shares that are assigned to the specified team project collection.
listscvmmlibraryshares	Lists all library shares that are available in VMM.

# Example

In this example, a library share is added to a team project collection.

### Copy Code

TFSLabConfig TPCLibraryShare
 /collection:http://abc:8080/TFS/Collection0
 /add
 /scvmmLibraryShare:"LibraryShares\LibraryShare1"
 /name:LibraryShare1

### See Also

### Reference

TFSLabConfig TPCHostGroup Command

### Concepts

Configuring Lab Management with TFSLabConfig Configuring and Administering Lab Management

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
Changing Groups and Permissions with TFSSecurity
Send Feedback

You can use the **TFSSecurity** command-line utility to create, modify, and delete groups and users in Visual Studio Team Foundation Server (TFS), in addition to modifying permissions for groups and users. For information about how to perform these tasks in the user interface, see <u>Configuring Users</u>, <u>Groups</u>, and <u>Permissions</u>.

By default, you can find this utility in Drive:\Program Files\Microsoft Team Foundation Server 11.0\Tools on the application-tier server or servers for Team Foundation.

#### ⊠Note

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

You can use the options in the following table for all commands in this utility.

Option	n Description
/?	Displays the command syntax and options for <b>TFSSecurity</b> .

# In This Section

**TFSSecurity Identity and Output Specifiers** 

/i Command

/im Command

/imx Command

/g Command

/gcg Command

/gc Command

/gun Command

/gud Command

/gd Command

/g+ Command

/g- Command

<u>/m Command</u>

/a+ Command

<u>/a- Command</u>

/acl Command

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSSecurity Identity and Output Specifiers
See Also Send Feedback

The input and output for the **TFSSecurity** command-line utility follows a standard format. The tables later in this topic describe valid identity and output specifiers for this command. These specifiers apply to all of the TFSSecurity command-line utilities.

### ⊠Note

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

# **Identity Specifiers**

You can reference an identity by using one of the notations in the following table.

Identity specifier	Description	Example
<b>sid:</b> Sid.	References the identity that has the specified security identifier (SID).	sid:S-1-5-21-2127521184-1604012920- 1887927527-588340
	References the identity that has the specified name. For Windows, Name is the account name. If the	
	referenced identity is in a domain, the domain name is	To reference the identity of the user "John Peoples" in the domain "Datum1" at the fictitious company "A. Datum Corporation:"
	required. For application	n:DATUM1\jpeoples
<b>n:</b> [Domain\]Name	groups, Name is the group	To reference application groups:
	display name, and Domain is	n:"Full-time Employees"
	the URI or GUID of the containing	n:00a10d23-7d45-4439-981b- d3b3e0b0b1ee\Vendors

		project. In this context, if Domain is omitted, the scope is assumed to be at the collection level.	
adm:[	Scope]	References the administrative application group for the scope, such as "Team Foundation Administrators" for the server level or "Project Collection Administrators" at the collection level. The optional parameter Scope is a project URI or URL, including its GUID and connection string. If scope is omitted, the server or collection scope is assumed based on whether the	adm:vstfs:///Classification/TeamProject/Gl

	/instance or /server parameter is used. In either case, the colon is still required.	
srv:	References the application group for service accounts.	Not applicable
all:	References all groups and identities.	Not applicable
String	References an unqualified string. If String starts with <b>S-1</b> -, it is identified as a SID. If String starts with <b>CN=</b> or <b>LDAP:</b> // it is identified as a distinguished name. Otherwise, String is identified as a name.	"Team testers"

# **Type Markers**

### **Identity Type Markers**

The following table lists identity type markers that are used in output messages.

Identity type marker	Description
U	Windows user.
G	Windows group.
Α	Team Foundation Server (TFS) application group.
a [A]	Administrative application group.
s [A]	Service account application group.
X	Identity is not valid.
?	Identity is unknown.

### ■Access Control Entry Markers

The following table lists access control entry markers that are used in output messages.

Access control entry marker	Description
+	ALLOW access control entry.
-	DENY access control entry.
* []	Inherited access control entry.

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
/i Command
See Also Send Feedback

Use /i to display identity information for a specified group in a deployment of Team Foundation Server.

**Required Permissions** 

To use the /i command, you must have the View collection-level information or the View instance-level information permission set to Allow, depending on whether you are using the /collection or /server parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /i Identity [/collection:CollectionURL] [/server:Serverl

### **Parameters**

Argument	Description
Identity	The identity of the user or the application group. For information about identity specifiers, see <u>TFSSecurit</u> and <u>Output Specifiers</u> .
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

The **/i** command of the **TFSSecurity** command-line utility displays information about each group within the team project collection (/server) or the application-tier server (/instance). It does not display any membership information.

# **Examples**

The following example displays identity information for the "Team Foundation Administrators" group.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

>tfssecurity /i "Team Foundation Administrators" /server:ServerURL

Sample output:

#### Copy Code

Resolving identity "Team Foundation Administrators"...

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Server scope Display name: Team Foundation Administrators Description: Members of this application group can perform all prime Description: Members of this application group can perform all prime Description: Members of this application group can perform all prime Description: Members of this application group can perform all prime Description: Members of this application group can perform all prime Description

The following example displays identity information for the Project Collection Administrators group using the adm: identity specifier.

### Copy Code

>tfssecurity /i adm: /collection:CollectionURL

Sample output:

Copy Cod

The following example displays identity information for the Project Administrators group for the "Datum" project by using the adm: identity specifier.

#### Copy Code

>tfssecurity /i adm:vstfs:///Classification/TeamProject/ProjectGUID

Sample output:

Copy Code

Resolving identity "adm:vstfs:///Classification/TeamProject/Project(

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Datum Display name: [Datum]\Project Administrators Description: Members of this application group can perform all ope

## See Also

### **Other Resources**

<u>Changing Groups and Permissions with TFSSecurity</u> How to: Create a Global Group Team Project Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /im Command
See Also Send Feedback

Use **/im** to display information about the identities that compose the direct membership of a group that you specify.

**Required Permissions** 

To use the **/im** command, you must have the View collection-level information or the View instance-level information permission set to Allow, depending on whether you are using the /collection or /server parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /im Identity [/collection:CollectionURL] [/server:Server

### **Parameters**

Argument	Description
Identity	The identity of the user or the group. For more inform about identity specifiers, see <u>TFSSecurity Identity an</u> <u>Specifiers</u> .
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName
# **Remarks**

Run this command on an application-tier server for Team Foundation.

The **/im** command of **TFSSecurity** displays the direct members of the specified group only. This list includes other groups that are members of the specified group. However, the actual members of the member groups are not listed.

## **Examples**

The following example displays direct membership identity information for the "Team Foundation Administrators" group in the domain "Datum1" at the fictitious company "A. Datum Corporation".

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

#### Copy Code

>tfssecurity /im "Team Foundation Administrators" /server:ServerURL

Sample output:

### Copy Code

Resolving identity "Team Foundation Administrators"...

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Server scope Display name: Team Foundation Administrators Description: Members of this application group can perform all privi

3 member(s):

- [U] Datum1\hholt (Holt, Holly)
- [G] BUILTIN\Administrators (BUILTIN\Administrators)
- s [A] [InstanceName]\Team Foundation Service Accounts

Member of 2 group(s):

- a [A] [DatumOne]\Project Collection Administrators ([DatumOne]\Proje
- e [A] [InstanceName]\Team Foundation Valid Users

Done.

The following example displays identity information for the Project Collection Administrators group in the "DatumOne" team project collection in the domain "Datum1" at the fictitious company "A. Datum Corporation" by using the adm: identity specifier.

#### Copy Code

>tfssecurity /im adm: /collection:CollectionURL

Sample output:

Copy Code

Resolving identity "adm: "...

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Server scope Display name: [DatumOne]\Project Collection Administrators Description: Members of this application group can perform all privi

```
5 member(s):
```

- [U] Datum1\jpeoples (Peoples, John)
- [U] Datum1\hholt (Holt, Holly)
- [G] BUILTIN\Administrators (BUILTIN\Administrators)
- a [A] [InstanceName]\Team Foundation Administrators

```
s [A] [DatumOne]\Project Collection Service Accounts ([DatumOne]\Pro
```

Member of 1 group(s):
e [A] [DatumOne]\Project Collection Valid Users ([DatumOne]\Project

Done.

The following example displays identity information for the Project Administrators group for the "Datum" project in the "DatumOne" team project collection in the domain "Datum1" at the fictitious company "A. Datum Corporation" using the adm: identity specifier.

#### Copy Code

>tfssecurity /im adm:vstfs:///Classification/TeamProject/ProjectGUII

Sample output:

#### Copy Code

Resolving identity "adm:vstfs:///Classification/TeamProject/Project(

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Datum Display name: [Datum]\Project Administrators Description: Members of this application group can perform all opera

2 member(s):
 [U] Datum1\jpeoples (Peoples, John)
 [U] Datum1\hholt (Holt, Holly)

Member of 1 group(s):
e [A] [DatumOne]\Project Collection Valid Users ([DatumOne]\Project

Done.

## See Also

### **Other Resources**

<u>Changing Groups and Permissions with TFSSecurity</u> How to: Create a Global Group Team Project Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /imx Command
See Also Send Feedback

Use **/imx** to display information about the identities that compose the expanded membership of a specified group.

**Required Permissions** 

To use the **/imx** command, you must have the View collection-level information or the View instance-level information permission set to Allow, depending on whether you are using the **/collection** or **/server** parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /imx Identity [/collection:CollectionURL] [/server:Serve

## **Parameters**

Argument	Description
Identity	The identity of the user or the group. For more inform about identity specifiers, see <u>TFSSecurity Identity an</u> <u>Specifiers</u> .
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

The **/imx** command of **TFSSecurity** displays the expanded members of the specified group only. This list includes not only other groups that are members of the specified group but also the members of the member groups.

## **Examples**

The following example displays expanded membership identity information for the "Team Foundation Administrators" group in the domain "Datum1" at the fictitious company "A. Datum Corporation".

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

#### Copy Code

>tfssecurity /imx "Team Foundation Administrators" /server:ServerURI

Sample output:

#### Copy Code

Resolving identity "Team Foundation Administrators"...

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Server scope Display name: Team Foundation Administrators Description: Members of this application group can perform all privi

10 member(s):

- [U] Datum1\hholt (Holly Holt)
- [U] Datum1\jpeoples (John Peoples)
- [U] Datum1\tommyh (Tommy Hartono)
- [U] Datum1\henriea (Henriette Andersen)
- [U] Datum1\djayne (Darcy Jayne)
- [U] Datum1\aprilr (April Reagan)
- [G] Datum1\InfoSec Secure Environment
- [U] Datum1\nbento (Nuno Bento)
- [U] Datum1\cristp (Cristian Petculescu)

- [G] BUILTIN\Administrators (BUILTIN\Administrators)
- s [A] [InstanceName]\Team Foundation Service Accounts

#### Member of 3 group(s):

- a [A] [DatumOne]\Project Collection Administrators ([DatumOne]\Proje
- e [A] [DatumOne]\Project Collection Valid Users ([DatumOne]\Project
- e [A] [InstanceName]\Team Foundation Valid Users

Done.

The following example displays identity information for the Project Collection Administrators group in the "DatumOne" team project collection in the domain "Datum1" at the fictitious company "A. Datum Corporation" using the adm: identity specifier.

#### Copy Code

>tfssecurity /imx adm: /collection:CollectionURL

Sample output:

#### Copy Code

Resolving identity "adm: "...

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Server scope Display name: [DatumOne]\Project Collection Administrators Description: Members of this application group can perform all privi

6 member(s):

- [U] Datum1\jpeoples (Peoples, John)
- [U] Datum1\hholt (Holt, Holly)
- [G] BUILTIN\Administrators (BUILTIN\Administrators)
- a [A] [InstanceName]\Team Foundation Administrators
- s [A] [InstanceName]\Team Foundation Service Accounts
- s [A] [DatumOne]\Project Collection Service Accounts ([DatumOne]\Pro

Member of 1 group(s):

e [A] [DatumOne]\Project Collection Valid Users ([DatumOne]\Project Done.

The following example displays identity information for the Project Administrators group for the "Datum" project in the "DatumOne" team project collection in the domain "Datum1" at the fictitious company "A. Datum Corporation" using the adm: identity specifier.

### Copy Code

>tfssecurity /imx adm:vstfs:///Classification/TeamProject/ProjectGU]

Sample output:

#### Copy Code

Resolving identity "adm:vstfs:///Classification/TeamProject/Project(

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Datum Display name: [Datum]\Project Administrators Description: Members of this application group can perform all opera

2 member(s):
 [U] Datum1\jpeoples (Peoples, John)
 [U] Datum1\hholt (Holt, Holly)

Member of 2 group(s):
e [A] [DatumOne]\Project Collection Valid Users ([DatumOne]\Project
e [A] [InstanceName]\Team Foundation Valid Users

Done.

For more information about the output specifiers, such as [G] and [U], see <u>TFSSecurity Identity and Output Specifiers</u>.

## See Also

### **Other Resources**

<u>Changing Groups and Permissions with TFSSecurity</u> How to: Create a Global Group Team Project Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /g Command
Example See Also Send Feedback

Use /**g** to list the groups in a team project, in a team project collection, or across Team Foundation Server.

**Required Permissions** 

To use the /**g** command, you must have the View collection-level information or the View instance-level information permission set to Allow, depending on whether you are using the /**collection** or /**server** parameter, respectively. To use the /g command within the scope of a single team project, you must have the View project-level information permissions set to Allow. For more information, see <u>Team Foundation Server Permissions</u>.

```
TFSSecurity /g [scope] [/collection:CollectionURL] [/server:ServerUF
```

## **Parameters**

Argument	Description
scope	Optional. Specifies the URI of the team project for w want to display groups. To obtain the URI for a team open Team Explorer, right-click the team project, clic Properties, and copy the entire entry for URL.
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

The /**g** command of the **TFSSecurity** command-line utility displays information about every group within the selected scope. This scope can be the team project collection (/**server**) or the application-tier server (/**instance**). If used with the scope of a team project, it will display information only about the groups associated with that team project.

# Example

The following example displays information for all the groups within a team project collection.

### Copy Code

>TFSSecurity /g /collection:CollectionURL

## See Also

### **Other Resources**

<u>Changing Groups and Permissions with TFSSecurity</u> How to: Create a Global Group Team Project Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /gcg Command
Example See Also Send Feedback

Use the /**gcg** command to create a server-level or collection-level group. To create a server-level or collection-level group from the user interface, see <u>Configuring Users, Groups, and Permissions</u>.

**Required Permissions** 

To use the **/gcg** command, you must have the Edit project-level information permission for that team project set to Allow. For more information, see <u>Team</u> <u>Foundation Server Permissions</u>.

TFSSecurity /gcg GroupName [GroupDescription] [/collection:Collectic

## **Parameters**

Argument	Description
GroupName	The group name.
GroupDescription	A description of the group. Optional.
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName <b>:</b> Port/VirtualDirectoryName

## **Remarks**

Run this command on an application-tier server for Team Foundation.

Server-level groups are created directly on the application tier and apply to all team project collections. Collection-level are created at the team project collection level. They apply to that collection and have implications for all team projects within the collection. In contrast, team project groups apply to a specific project within a collection but not any other projects in that collection. You can assign permissions to server-level groups so that members of those groups can perform tasks in Team Foundation Server (TFS) itself, such as creating team project collections. You can assign permissions to collection-level groups so that members of those groups so that members of those groups can perform tasks across a team project collection, such as administering users.

### ⊠Note

You can use the **/gcg** command to create groups, but you cannot use it to add any users to the groups or assign any permissions. For information about how to change the membership of a group, see <u>/g+ Command</u> and <u>/g- Command</u>. For information about how to change the permissions for the group, see <u>/a+</u> <u>Command</u> and <u>/a- Command</u>.

# Example

The following example creates a collection-level group that is named "Datum Testers" with the description "A. Datum Corporation Testers."

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

#### Copy Code

>TFSSecurity /gcg "Datum Testers" "A. Datum Corporation Testers" /cc

The following example creates a server-level group that is named "Datum Auditors" with the description "A. Datum Corporation Auditors."

### Copy Code

>TFSSecurity /gcg "Datum Auditors" "A. Datum Corporation Auditors" /

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /gc Command
Example See Also Send Feedback

Use /**gc** at a command prompt to create a project-level group. To create a project-level group from the user interface, see <u>Configuring Users</u>, <u>Groups</u>, <u>and</u> <u>Permissions</u>.

**Required Permissions** 

To use the /**gc** command, you must have the Edit Project-Level Information permission for that team project set to Allow. For more information, see <u>Team</u> <u>Foundation Server Permissions</u>.

TFSSecurity /gc Scope GroupName [GroupDescription] [/collection:Coll

## **Parameters**

Argument	Description
Scope	The URI of the team project to which you want to ad- level group. To obtain the URI for a team project, cor and open Team Explorer, hover over the name of the Home, and read the address. Alternatively, connect to project in Web Access and copy the URL.
GroupName	The name of the new group.
GroupDescription	A description of the project group. Optional.
/ <b>collection</b> :CollectionURL	The URL of the team project collection. Required. The will be created within the team project collection. The for the URL is http://ServerName:Port/VirtualDirectoryName/Colle

# **Remarks**

Run this command on an application-tier server for Team Foundation.

A project-level group is a security group for your team project. You can use project groups to grant read, write, and administrative permissions that meet the security requirements of your organization.

# Example

The following example creates a group that is specific to the project that the URI "vstfs://Classification/TeamProject/00000000-0000-0000-0000-00000000000" specifies. The group is named "Test Group" and has the description "This group is for testing."

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

You must replace the placeholder GUID with the URI of the team project for which you want to create this group. To obtain the URI for a team project, open Team Explorer, right-click the team project, click Properties, and copy the entire value of the URL property.

After you run the command, you can verify the group in Team Explorer. Rightclick the team project that you used in the command, click Team Project Settings, and then click Group Memberships. In the Project Groups on TeamProjectName dialog box, the Groups list includes Test Group .

#### ⊠Note

You can use the /**gc** command to create groups but not to add any users to the groups or assign any permissions. To change the membership of the group, see /**g**+ Command and /**g**- Command. To change the permissions for the group, see /**a**+ Command and /**a**- Command.

>TFSSecurity /gc "vstfs:///Classification/TeamProject/00000000-0000.

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /gun Command
Example See Also Send Feedback

Use /**gun** to rename a server-level or collection-level group.

**Required Permissions** 

To use the **/gun** command, you must have the View collection-level information and Edit collection-level information or the View instance-level information and Edit instance-level information permissions set to Allow, depending on whether you are using the **/collection** or **/server** parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /gun GroupIdentity GroupName [/collection:CollectionURL]

## **Parameters**

Argument	Description
GroupIdentity	Specifies the group identity. For more information ab identity specifiers, see <u>TFSSecurity Identity and Outr</u> <u>Specifiers</u> .
GroupName	Specifies the new name of the group.
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: http://ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URI application-tier server in the following format: http://ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

# Example

The following example renames the collection-level group "A. Datum Corporation Testers" to "A. Datum Corporation Test Engineers."

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

>tfssecurity /gun "A. Datum Corporation Testers" "A. Datum Corporati

### See Also

### Concepts

**TFSSecurity Identity and Output Specifiers** 

#### **Other Resources**

Changing Groups and Permissions with TFSSecurity How To: Rename a Team Project Group How to: Rename a Global Group

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
/gud Command
Example See Also Send Feedback

Use /**gud** to change the description for a server-level or collection-level group.

**Required Permissions** 

To use the /**gud** command, you must have the Edit project-level information permission set to Allow. For more information, see <u>Team Foundation Server</u> <u>Permissions</u>.

TFSSecurity /gud GroupIdentity GroupDescription [/collection:Collect

## **Parameters**

Argument	Description
GroupIdentity	Specifies the group identity. For more information ab identity specifiers, see <u>TFSSecurity Identity and Outr</u> <u>Specifiers</u> .
GroupDescription	Specifies the new description for the group.
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: http://ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URI application-tier server in the following format: http://ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

# Example

The following example associates the description "The members of this group test the code for this project" with the group "Datum Testers."

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Cod
## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

Managing Groups Default Groups Global Groups Team Project Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /gd Command
Example See Also Send Feedback

Use /**gd** to delete a server-level or collection-level group.

**Required Permissions** 

To use the **/gd** command, you must have the View collection-level information and Edit collection-level information or the View instance-level information and Edit instance-level information permissions set to Allow, depending on whether you are using the **/collection** or **/server** parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /gd groupIdentity [/collection:CollectionURL] [/server:

## **Parameters**

Argument	Description
groupIdentity	Specifies the group identity. For more information ab identity specifiers, see <u>Changing Groups and Permiss</u> <u>TFSSecurity</u> .
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

You can also remove groups on Team Explorer. For more information, see How to: Remove a Global Group and How To: Remove a Team Project Group.

# Example

The following example deletes a group from the team project collection. The group is identified by "S-1-5-21-2127521184-1604012920-1887927527-588340", the security identifier (SID). For more information about finding the SID of a group, see <u>/im Command</u>. You can also use the friendly name to delete a group.

#### ĭNote

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

>TFSSecurity /gd S-1-5-21-2127521184-1604012920-1887927527-588340 /(

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

How to: Create a Global Group Team Project Groups Default Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /g+ Command
See Also Send Feedback

Use /**g**+ to add a user or another group to an existing group.

**Required Permissions** 

To use the /**g**+ command, you must have the View collection-level information and Edit collection-level information or the View instance-level information and Edit instance-level information permissions set to Allow, depending on whether you are using the /collection or /server parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /g+ groupIdentity memberIdentity [/collection:Collection

## **Parameters**

Argument	Description
groupIdentity	Specifies the group identity. For more information on identity specifiers, see <u>TFSSecurity Identity and Outr</u> <u>Specifiers</u> .
memberIdentity	Specifies the member identity. For more information identity specifiers, see <u>TFSSecurity Identity and Outr</u> <u>Specifiers</u> .
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: http://ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

You can also add users and groups to an existing group using Team Explorer. For more information, see How to: Add Users to a Global Group.

# **Examples**

The following example adds the Datum1 domain user John Peoples (Datum1\jpeoples) to the Team Foundation Administrators group.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

#### Copy Code

>tfssecurity /g+ "Team Foundation Administrators" n:Datum1\jpeoples

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved.

The target Team Foundation Server is http://ADatumCorporation:8080/ Resolving identity "Team Foundation Administrators"... a [A] [INSTANCE]\Team Foundation Administrators Resolving identity "n:Datum1\jpeoples"...

[U] DATUM1\jpeoples (John Peoples) Adding John Peoples to [INSTANCE]\Team Foundation Administrators... Verifying...

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Server scope Display name: [INSTANCE]\Team Foundation Administrators Description: Members of this group can perform all operations on 1 4 member(s):

[U] Datum1\hholt (Holly Holt)

- [U] Datum1\jpeoples (John Peoples)
  [G] BUILTIN\Administrators (BUILTIN\Administrators)
- s [A] [INSTANCE]\Team Foundation Service Accounts

Member of 2 group(s):

- a [A] [Collection0] Project Collection Administrators
- e [A] [INSTANCE]\Team Foundation Valid Users

Done.

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

How to: Create a Global Group Team Project Groups Default Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management
/g- Command
See Also Send Feedback

Use /**g**- to remove a user or a user group from an existing group.

**Required Permissions** 

To use the **/g-** command, you must have the View collection-level information and Edit collection-level information or the View instance-level information and Edit instance-level information permissions set to Allow, depending on whether you are using the **/collection** or **/server** parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /g- groupIdentity memberIdentity [/collection:Collection

## **Parameters**

Argument	Description
groupIdentity	Specifies the group identity. For more information ab identity specifiers, see <u>TFSSecurity Identity and Outr</u> <u>Specifiers</u> .
memberIdentity	Specifies the member identity. For more information valid identity specifiers, see <u>TFSSecurity Identity and</u> <u>Specifiers</u> .
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName <b>:</b> Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

You can also add users and groups to an existing group using Team Explorer. For more information, see How to: Remove Users from a Team Project Group, How to: Remove Users from a Default Group or How to: Remove Users from a Server-Level Group.

# **Examples**

The following example removes the Datum1 domain user John Peoples (Datum1\jpeoples) from the Team Foundation Administrators group.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

#### Copy Code

>tfssecurity /g- "Team Foundation Administrators" n:Datum1\jpeoples

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved.

The target Team Foundation Server is http://ADatumCorporation:8080/ Resolving identity "Team Foundation Administrators"... a [A] [INSTANCE]\Team Foundation Administrators Resolving identity "n:Datum1\jpeoples"...

[U] DATUM1\jpeoples (John Peoples)
Removing John Peoples from [INSTANCE]\Team Foundation Administrators
Verifying...

DN:

Identity type: Team Foundation Server application group Group type: AdministrativeApplicationGroup Project scope: Server scope Display name: [INSTANCE]\Team Foundation Administrators Description: Members of this group can perform all operations on 1 3 member(s):

[U] Datum1\hholt (Holly Holt)

- [G] BUILTIN\Administrators (BUILTIN\Administrators)
  s [A] [INSTANCE]\Team Foundation Service Accounts

#### Member of 2 group(s):

- a [A] [Collection0]\Project Collection Administrators e [A] [INSTANCE]\Team Foundation Valid Users

Done.

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /m Command
See Also Send Feedback

Use /**m** to check explicit and implicit group membership information for a specified group or user.

**Required Permissions** 

To use the **/m** command, you must be a member of the Team Foundation Administrators security group. For more information, see <u>Team Foundation</u> <u>Server Permissions</u>.

#### ĭNote

Even if you are logged on with administrative credentials, you must open an elevated Command Prompt to perform this function.

TFSSecurity /m GroupIdentity [MemberIdentity] [/collection:Collectic

## **Parameters**

Argument	Description
GroupIdentity	Specifies the group identity. For more information on identity specifiers, see <u>TFSSecurity Identity and Out</u> <u>specifiers</u> .
MemberIdentity	Specifies the member identity. By default, the value c argument is the identity of the user who is running the command. For more information on valid identity spe see <u>TFSSecurity Identity and Output Specifiers</u> .
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URI application-tier server in the following format: http://ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on the local application-tier computer.

The **/m** command of the **TFSSecurity** command-line utility checks both direct and extended memberships.

# **Examples**

The following example verifies whether the user "Datum1\jpeoples" belongs to the Team Foundation Administrators server-level group.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

#### Copy Code

>TFSSecurity /m "Team Foundation Administrators" n:Datum1\jpeoples /

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved.

The target Team Foundation Server is http://ADatumCorporation:8080/ Resolving identity "Team Foundation Administrators"... a [A] [INSTANCE]\Team Foundation Administrators Resolving identity "n:Datum1\jpeoples"... [U] DATUM1\jpeoples (John Peoples)

Checking group membership...

John Peoples IS a member of [INSTANCE]\Team Foundation Administrator

Done.

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /a+ Command
See Also Send Feedback

Use /**a**+ to add permissions for a user or a group in a server-level, collectionlevel, or project-level group. To add users to groups from the user interface, see <u>Configuring Users, Groups, and Permissions</u>.

**Required Permissions** 

To use the /**a**+ command, you must have the View collection-level information or the View instance-level information permission set to Allow, depending on whether you are using the /**collection** or /**server** parameter, respectively. If you are changing permissions for a team project, you must also have the Edit projectlevel information permission for the team project set to Allow. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /a+ Namespace Token Action Identity (ALLOW | DENY) [/co]

## **Parameters**

Argument	Description
Namespace	The namespace that contains the group to which you add permissions for a user or group. You can also use <b>TFSSecurity</b> / <b>a</b> command to view a list of namespace server, collection, and project level.
	The name or GUID of the object on which you want permissions.
Token	⊠Note
	Tokens vary depending on the namespace you specify namespaces do not have tokens that apply for this co
Action	The name of the permission for which you are grantin denying access. For a list of valid IDs, see <u>Team Foun</u> <u>Server Permissions</u> , or use the <b>TFSSecurity</b> / <b>a</b> comm view a list of valid actions for a namespace that you s
	The identity of the user or the group. For more inform about identity specifiers, see <u>TFSSecurity Identity an</u> <u>Specifiers</u> .
	• ALLOW
Identity	The group or user can perform the operation that Action specifies.
	• DENY

	The group or user cannot perform the operation Action specifies.
/ <b>collection:</b> CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName <b>:</b> Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

Access control entries are security mechanisms that determine which operations a user, group, service, or computer is authorized to perform.

# **Examples**

The following example displays what namespaces are available at the server level for the application-tier server that is named ADatumCorporation.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

#### Copy Code

>tfssecurity /a /server:ServerURL

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved.

The target Team Foundation Server is http://ADatumCorporation:8080/.

The following security namespaces are available to have permissions

Registry Identity Job Server CollectionManagement Warehouse Catalog EventSubscription Lab

Done.

The following example displays what actions are available for the Server namespace at the collection level.

#### Copy Code

>tfssecurity /a Server /collection:CollectionURL

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved. The target Team Foundation Server is http://ADatumCorporation:8080/ The following actions are available in the security namespace Server GenericRead GenericWrite Impersonate TriggerEvent

Done.

The following example grants the server-level "View instance-level information" permission to the ADatumCorporation deployment for the Datum1 domain user John Peoples (Datum1\jpeoples).

>tfssecurity /a+ Server FrameworkGlobalSecurity GenericRead n:Datum:

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved. The target Team Foundation Server is http://ADatumCorporation:8080/ Resolving identity "n:Datum1\jpeoples"... [U] Datum1\jpeoples (John Peoples) Adding the access control entry... Verifying... Effective ACL on object "FrameworkGlobalSecurity": [+] GenericRead [INSTANCE]\Team Foundation [+] GenericRead [INSTANCE]\SharePoint Web / [+] Impersonate [INSTANCE]\SharePoint Web / [+] GenericRead [INSTANCE]\Team Foundation [+] GenericWrite [INSTANCE]\Team Foundation [INSTANCE]\Team Foundation [+] Impersonate [+] TriggerEvent [INSTANCE]\Team Foundation [+] GenericRead [INSTANCE]\Team Foundation [INSTANCE]\Team Foundation [+] GenericWrite [INSTANCE]\Team Foundation [+] TriggerEvent [+] GenericRead DATUM1\jpeoples

Done.

The following example grants the collection-level "View collection-level information" permission to the Collection0 team project collection for Datum1 domain user John Peoples (Datum1\jpeoples).

>tfssecurity /a+ Server FrameworkGlobalSecurity GenericRead n:Datum1

Sample output:

Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved. The target Team Foundation Server is http://ADatumCorporation:8080/( Resolving identity "n:Datum1\jpeoples"... [U] DATUM1\jpeoples (John Peoples) Adding the access control entry... Verifying... Effective ACL on object "FrameworkGlobalSecurity": [Collection0]\Project Colle [+] GenericRead [+] GenericRead [Collection0]\Project Colle [Collection0]\Project Colle [+] GenericWrite [Collection0]\Project Colle [+] Impersonate [+] TriggerEvent [Collection0]\Project Colle [Collection0]\Project Colle [+] GenericRead [+] GenericWrite [Collection0]\Project Colle [Collection0]\Project Colle [+] TriggerEvent

[+] GenericRead
[+] Impersonate
[+] GenericRead
[+] GenericRead

Done.

[INSTANCE]\SharePoint Web /
[INSTANCE]\SharePoint Web /
[Collection0]\Project Coll(
DATUM1\jpeoples

## See Also

### **Other Resources**

Changing Groups and Permissions with TFSSecurity

How to: Create a Global Group Team Project Groups Default Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /a- Command
See Also Send Feedback

Use the **/a-** command to remove a user or a group from membership in a serverlevel, collection-level, or project-level group. To add users to groups from the user interface, see <u>Configuring Users</u>, <u>Groups</u>, <u>and Permissions</u>.

**Required Permissions** 

To use the **/a-** command, you must have the View collection-level information or the View instance-level information permission set to Allow, depending on whether you are using the **/collection** or **/server** parameter, respectively. If you are changing permissions for a team project, you must also have the Edit project-level information permission for the team project set to Allow.

TFSSecurity /a- Namespace Token Action Identity (ALLOW | DENY) [/co]

## **Parameters**

Argument	Description
Namespace	The namespace that contains the group from which yeremove the user or group. You can also use the <b>TFSS</b> command to view a list of namespaces at the server le collection level, and the project level.
	The name or GUID of the object on which you want permissions.
Token	⊠Note
	Tokens vary depending on the namespace that you sp Some namespaces do not have tokens that apply for t command.
Action	The name of the permission that for which access is $\xi$ denied. For a list of valid IDs, see <u>Team Foundation </u> <u>Permissions</u> , or use the <b>TFSSecurity</b> / <b>a</b> command to of valid actions for a namespace that you specify.
	The identity of the user or the group. For more inform about the identity specifiers, see <u>TFSSecurity Identity</u> <u>Output Specifiers</u> .
	• ALLOW
Identity	The group or user can perform the operation that Action specifies.

	• DENY
	The group or user cannot perform the operation Action specifies.
/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: http://ServerName:Port/VirtualDirectoryName/Colle
/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URL application-tier server in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

Access control entries are security mechanisms that determine which operations a user, group, service, or computer is authorized to perform on a computer or server.
# **Examples**

The following example displays what namespaces are available at the server level for the application-tier server that is named ADatumCorporation.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

>tfssecurity /a /server:ServerURL

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved.

The target Team Foundation Server is http://ADatumCorporation:8080/.

The following security namespaces are available to have permissions

Registry Identity Job Server CollectionManagement Warehouse Catalog EventSubscription Lab

Done.

The following example displays what actions are available for the Server namespace at the collection level.

#### Copy Code

>tfssecurity /a Server /collection:CollectionURL

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved. The target Team Foundation Server is http://ADatumCorporation:8080/ The following actions are available in the security namespace Server GenericRead GenericWrite Impersonate TriggerEvent

Done.

The following example removes the server-level "View instance-level information" permission to the ADatumCorporation deployment for the Datum1 domain user John Peoples (Datum1\jpeoples).

## 

>tfssecurity /a- Server FrameworkGlobalSecurity GenericRead n:Datum:

Sample output:

### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved. The target Team Foundation Server is http://ADatumCorporation:8080/ Resolving identity "n:Datum1\jpeoples"... [U] Datum1\jpeoples (John Peoples) Removing the access control entry...

Verifying...

Effective ACL on object "FrameworkGlobalSecurity": [+] GenericRead [INSTANCE]\Team Foundation [+] GenericRead [INSTANCE]\SharePoint Web / [+] Impersonate [INSTANCE]\SharePoint Web / [+] GenericRead [INSTANCE]\Team Foundation [+] GenericWrite [INSTANCE]\Team Foundation [INSTANCE]\Team Foundation [+] Impersonate [+] TriggerEvent [INSTANCE]\Team Foundation [+] GenericRead [INSTANCE]\Team Foundation [+] GenericWrite [INSTANCE]\Team Foundation [INSTANCE]\Team Foundation [+] TriggerEvent

Done.

The following example removes the collection-level "View collection-level information" permission to the Collection0 team project collection for Datum1 domain user John Peoples (Datum1\jpeoples).

>tfssecurity /a+ Server FrameworkGlobalSecurity GenericRead n:Datum1

Sample output:

#### Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved. The target Team Foundation Server is http://ADatumCorporation:8080/( Resolving identity "n:Datum1\jpeoples"... [U] DATUM1\jpeoples (John Peoples) Removing the access control entry... Verifying... Effective ACL on object "FrameworkGlobalSecurity": [+] GenericRead [Collection0]\Project Coll6 [+] GenericRead [Collection0]\Project Coll6 [+] GenericWrite [Collection0]\Project Coll6

- [+] Impersonate
- [+] TriggerEvent
- [+] GenericRead
- [+] GenericWrite
- [+] TriggerEvent
  [+] GenericRead

[Collection0]\Project Coll [Collection0]\Project Coll [Collection0]\Project Coll [Collection0]\Project Coll [Collection0]\Project Coll [INSTANCE]\SharePoint Web / [+] Impersonate
[+] GenericRead

Done.

[INSTANCE]\SharePoint Web #
[Collection0]\Project Colle

# See Also

## **Other Resources**

Changing Groups and Permissions with TFSSecurity

How to: Create a Global Group Team Project Groups Default Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Visual Studio Application Lifecycle Management /acl Command
See Also Send Feedback

Use /**acl** to display the access control list that applies to a particular object.

**Required Permissions** 

To use the /**acl** command, you must have the View collection-level information or the View instance-level information permission set to Allow, depending on whether you are using the /**collection** or /**server** parameter, respectively. For more information, see <u>Team Foundation Server Permissions</u>.

TFSSecurity /acl Namespace Token [/collection:CollectionURL] [/serve

## **Parameters**

	Argument	Description
	Namespace	The namespace that contains the group for which you view permissions for a user or group.
	Token	The name or GUID of the object on which you want the permissions.
		⊠Note
		Tokens vary depending on the namespace that you sp Some namespaces do not have tokens that apply for t command.
	/ <b>collection</b> :CollectionURL	Required if / <b>server</b> is not used. Specifies the URL of project collection in the following format: <b>http:</b> //ServerName:Port/VirtualDirectoryName/Colle
	/ <b>server</b> :ServerURL	Required if / <b>collection</b> is not used. Specifies the URI application-tier server in the following format: http://ServerName:Port/VirtualDirectoryName

# **Remarks**

Run this command on an application-tier server for Team Foundation.

Access control entries are security mechanisms that determine which operations a user, group, service, or computer is authorized to perform on a computer or server.

# **Examples**

The following example displays what users and groups have access to the FrameworkGlobalSecurity token in the Server namespace within the ADatumCorporation deployment.

#### ⊠Note

The examples are for illustration only and are fictitious. No real association is intended or inferred.

### Copy Code

> tfssecurity /acl Server FrameworkGlobalSecurity /server:ServerURL

Sample output:

## Copy Code

TFSSecurity - Team Foundation Server Security Tool Copyright (c) Microsoft Corporation. All rights reserved. The target Team Foundation Server is http://ADatumCorporation:8080/ Retrieving the access control list for object "Server"...

Effective ACL on object "FrameworkGlobalSecurity":

[+] GenericRead	[INSTANCE]\Team Foundation
[+] GenericRead	[INSTANCE]\SharePoint Web /
[+] Impersonate	[INSTANCE]\SharePoint Web /
[+] GenericRead	[INSTANCE]\Team Foundation
[+] GenericWrite	[INSTANCE]\Team Foundation
[+] Impersonate	[INSTANCE]\Team Foundation
[+] TriggerEvent	[INSTANCE]\Team Foundation
[+] GenericRead	[INSTANCE]\Team Foundation
[+] GenericWrite	[INSTANCE]\Team Foundation
[+] TriggerEvent	[INSTANCE]\Team Foundation
[+] GenericRead	DATUM1\jpeoples

Done.

# See Also

## **Other Resources**

Changing Groups and Permissions with TFSSecurity

How to: Create a Global Group Team Project Groups Default Groups

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
TFSServiceControl Command
Example See Also Send Feedback

You can use the **TFSServiceControl** command to stop or start all of the services and application pools Visual Studio Team Foundation Server 2012 (TFS) uses. For example, you use this command when backing up or restoring databases, or when you are moving your deployment from one machine to another.

#### ⊠Note

You must use the **TFSServiceControl** command to ensure that all necessary operations, services, and application pools are stopped for maintenance tasks such as backup and restore. You cannot manually perform all of the tasks carried out by the **TFSServiceControl** command.

**Required Permissions** 

To use the **TFSServiceControl** command, you must be a member of the Team Foundation Administrators security group, a member of the Administrators group on the application-tier server, and a member of the sysadmin security group for any SQL Server databases that TFS uses. For more information, see <u>Team Foundation Server Permissions</u>.

#### ⊠Note

Even if you log on with administrative credentials, you must open an elevated Command Prompt window to perform this function.

TFSServiceControl [quiesce|unquiesce]

## **Parameters**

Option	Description
quiesce	Stops or pauses all of the services, application pools, and operations in your deployment of TFS. This is required for certain maintenance tasks, such as restoring databases.
unquiesce	Starts or restarts all of the services, application pools, and operations in your deployment of TFS. This is required to return your server to operation after you run the command with the <b>quiesce</b> option.

# **Remarks**

You use the **TFSServiceControl** command as part of specific maintenance tasks. After you specify the **quiesce** option, the server will not operate until you specify the **unquiesce** option. By default, the **TFSServiceControl** command is located in the %programfiles%\Microsoft Team Foundation Server 11.0\Tools directory.

# Example

The following example shows how to stop a deployment of Team Foundation Server.

## Copy Code

TFSServiceControl quiesce

The following example shows how to start a deployment of Team Foundation Server.

## Copy Code

TFSServiceControl unquiesce

## **See Also**

### Tasks

Restore Data to the Same Location

## Concepts

Backing up and Restoring Your Deployment The Team Foundation Administration Console

### **Other Resources**

Restoring a Single-Server Deployment to New Hardware [Tutorial]

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Delete a Team Project [TFSDeleteProject]
Example See Also Send Feedback

You can remove a team project from Team Foundation Server when the project is no longer required by using **TFSDeleteProject**. In addition, if there are components that remain undeleted after an unsuccessful team project creation, you can use **TFSDeleteProject** to remove them.

## **▲Important**

**TFSDeleteProject** permanently destroys the team project, after which it cannot be recovered. You should backup all important project data before using **TFSDeleteProject**.

To access the **TFSDeleteProject** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

## Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, click Start, right-click Command Prompt, and then click Run as Administrator. For more information, see the Microsoft Web site: <u>User Access Control</u>.

**Required Permissions** 

To use the **TFSDeleteProject** command, you must be a member of the Team Foundation Administrators security group or the Project Administrators security group. For more information, see <u>Team Foundation Server Permissions</u> and <u>Team Foundation Server Default Groups, Permissions, and Roles</u>.

TFSDeleteproject [/q] [/force] [/excludewss] /collection:URL TeamPro

# **Parameters**

Option	Description
/ <b>q</b>	Optional. Use the quiet mode. Do not prompt the user for confirmation.
/force	Optional. Specifies that the deletion process should continue even if some components cannot be deleted.
/excludewss	Optional. Specifies to not delete the SharePoint site that is associated with the team project. Specify this option to mainta the existing site so that other team projects can continue using it.
	Required. Specifies the URI of the team project collection. Yo must use the following format for the URI: http://ServerName:Port/VirtualDirectoryName/CollectionNar
/ <b>collection</b> :URL	If you do not specify a virtual directory, you must use the following format for the URI:
	http://ServerName:Port/CollectionName.
TeamProjectNam	Required. The name of the project. If the name includes space enclose it in quotations marks.

# **Remarks**

When you create a team project, Team Foundation Server creates data objects on the server that hosts Team Foundation Server, and may create data objects on the server that hosts SharePoint Products, and the server that hosts SQL Server Reporting Services. When you remove a team project, the reports are automatically removed from SQL Server Reporting Services.

When you remove a team project, you can choose whether or not to remove the objects that were created to support the SharePoint site. However, an error can prevent Team Foundation Server from creating or deleting all the objects. To troubleshoot these problems, the following sections provide background information, links to other resources, and specific steps that help you determine the cause of the problem, fix the problem, and when necessary delete data objects that remain after running **TFSDeleteProject**.

- <u>TFSDeleteProject Process</u>
- Data That May Remain Undeleted
- Verify Team Project Components Are Deleted
- <u>Remove Remaining Components After Partial Project Deletion</u>
- Increase the Time-Out Period

## **TFSDeleteProject Process**

When you use the **TFSDeleteProject** command-line tool, it first deletes project data and then project Web sites.

## **Phase 1: TFSDeleteProject Deletes Project Data**

In the first phase, **TFSDeleteProject** automatically performs the following steps to remove team project data:

1. TFSDeleteProject creates an inventory of all the components that are

candidates for deletion. This includes components that integrate with Test Manager, Team Foundation Build, and Team Foundation version control.

- 2. **TFSDeleteProject** deletes the component that displays the project node in Team Explorer.
- 3. **TFSDeleteProject** flags the version control information for deletion, but does not immediately delete this information. The information includes all version control branches in the specified project, but no other branches outside of the project.
  - If a parent branch and a child branch are both in the project, **TFSDeleteProject** flags both for deletion.
  - If parent and child branches are in different projects, **TFSDeleteProject** flags only the specified branch.
  - If another project is a branch from the specified project,
     **TFSDeleteProject** flags only the specified project. When the specified project is deleted, the branch project becomes an orphan.
- 4. **TFSDeleteProject** immediately deletes build data, including information and core data, build definitions, build agents, and test results associated with the team project. The tool does not delete build drop locations. You do not need to delete the build drop location of an old team project before creating a team project that uses the same build drop location.

If the specified project contains a large amount of build data, the deletion might not finish within the timeout period. To work around this problem, see

Increase the Time-Out Period, and then run **TFSDeleteProject** again.

• **TFSDeleteProject** immediately deletes work items and work item fields that belong to the specified project, and it deletes all non-shared metadata.

If the specified project contains a large amount of work item data, the deletion might not finish within the timeout period. To solve this problem, see <u>Increase</u> <u>the Time-Out Period</u>, and then run **TFSDeleteProject** again.

## **Phase 2: TFSDeleteProject Deletes Project Web Sites**

In the second phase, **TFSDeleteProject** deletes the following data:

## **▲Important**

These steps can take a long time to complete, and during that time they can degrade server performance.

- **TFSDeleteProject** Uses the Reporting Services APIs to delete reports on the server that hosts Reporting Services.
- **TFSDeleteProject** Deletes the project portal Web site from the server that hosts SharePoint Products. This step occurs only if the project owns the site and site deletion is not excluded in the command line. (consider that multiple projects may point to a single site, but only one of them can be the owner where reports/dashboards are by default using this project).

## ⊠Note

Prior to deleting a team project, you can confirm that Reporting Services and SharePoint Products are using the correct project URLs by verifying the portal settings. For more information, see How to: Add a Team Project Portal.

If **TFSDeleteProject** successfully deletes all of the above data elements, it returns the message Done. To verify this result, see <u>Verify Team Project</u> <u>Components Are Deleted</u>.

If one or more components are not removed, you can rerun **TFSProjectDelete** by using the **/force** option to continue the deletion process even if it is unable to delete all data elements. With this option **TFSDeleteProject**, skips a component that it cannot delete, returns an error message, deletes the next component, and leaves the team project metadata and security settings intact.

## **Data That May Remain Undeleted**

The following data might remain in the deployment after **TFSDeleteProject** successfully completes:

- Team project data in the cube. Team project data remains in the cube until the cube is rebuilt, at which time the warehouse controller service removes all of the historic build data that has been deleted from the Team Foundation databases.
- Build drop files and folders. Build binaries, build log files, and log files containing test results are published during the build process. The locations for these files are not deleted. If you want to remove these files, you must remove them manually.
- Work item tracking metadata that is shared. **TFSDeleteProject** does not delete any metadata for work item tracking that is shared between team projects.
- Version control shelvesets containing shared code. Version control shelvesets are not deleted if there is code in the shelveset from multiple team projects.

# **Verify Project Deletion**

You can verify the success of a project deletion by confirming that the team project node no longer appears in Team Explorer and that its project portal Web site and reports folders no longer exist.

## To verify that a team project has been successfully deleted

- 1. Open Team Explorer and verify that the project does not appear as a project node.
- 2. Open Internet Explorer and type the URL of the project portal Web site. Verify that the site no longer exists.
- 3. In Internet Explorer, in the Address box, type the URL of the Reporting Services Web site using one of the following URL formats:
  - http://ReportingServices/Reports
  - http://ReportingServices/Reports\_TFSInstance
- 4. In Report Manager, click Show Details.
- 5. Verify that the folder for the deleted team project no longer appears. Click the root folder TfsReports, and then click the folder named for the team project collection. There should no longer be a folder with the name of the deleted project.
- 6. If either the reports or the Web site remain, see the next procedure.

# **Remove Remaining Components After Partial Project Deletion**

If the project portal Web site and reports folder remain after you delete a team project, remove the site and folder manually.

# To manually remove reports and the project portal Web site of a deleted team project

- 1. Log on to the server that hosts Reporting Services for the team project that you deleted.
- 2. Open Internet Explorer, and in the Address box type the URL of the Reporting Services Web site using one of the following URL formats:
  - http://localhost/Reports
  - o http://localhost/Reports\_TFSInstance
- 3. In Report Manager, click Show Details.
- 4. Click the root folder TfsReports, and then click the folder named for the team project collection.
- 5. Select the check box for the team project that was deleted.
- 6. Click Delete.
- 7. Click OK to confirm that you want to delete the reports folder for the team project.
- 8. To remove the project portal Web site of a deleted team project, see the following page on the Microsoft Web site:

How to: Create, Edit, and Delete Windows SharePoint Services Sites.

# **Increase the Time-Out Period**

By default, each Web service call that the **TFSDeleteProject** command issues to delete a component must complete within 10 minutes. If there are six calls, then the process could take up to an hour. If you want to delete a team project that is associated with a large amount of data, you can temporarily increase this time-out period.

### ⊠Note

When you increase the time-out period, the change affects all Web service calls. In general, you want to keep the time-out period to within 10 minutes to prevent Web service calls from degrading server performance and causing users to be locked out from using the user interface for long periods of time. Therefore, after the project is successfully deleted, you should change the time-out period back to 10 minutes.

### **Required Permissions**

To complete these procedures, you must be a Windows Administrator on the application-tier server.

### **▲Important**

Improperly modifying your computer's registry can cause your computer to become unstable. If you are not familiar with the registry, you should not add or remove entries, or modify it in any way.

# To increase the time-out period for the TFSDeleteProject command

- 1. Log on to the application-tier server.
- 2. Click Start, click Run, type regedit, and then click OK.

- 3. In the browser pane, expand **HKEY\_LOCAL\_MACHINE**:
  - If the server runs a 32-bit operating system, expand: HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\VisualStudio\11.
  - If the server runs a 64-bit operating system, expand: HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432\Microsoft\Visua
- 4. If the TeamFoundation\RequestSettings key does not exist, follow these steps to create it:
  - 1. Right-click TeamFoundation, point to New, and click Key.
  - 2. Name the key RequestSettings.
  - 3. Right-click RequestSettings, point to New, and click DWORD Value.
  - 4. Name the new value DefaultTimeout.
- 5. Right-click DefaultTimeout and choose Modify.
- 6. In Value Data, type the time-out period in milliseconds, and then click Decimal.

For example, to increase the time-out period to 30 minutes, type 1800000. To change the time-out period back to 10 minutes, type 600000.

- 7. Click OK.
- 8. On the File menu, click Exit.

# Example

The following command removes all components associated with the team project StoreFront on the Team Foundation Server AdventureWorks1 server in project collection Collection1 and from Team Explorer.

Copy Code

TFSDeleteProject /force /collection:http://AdventureWorks1:8080/tfs,

## See Also

### Tasks

Create a Team Project

## Concepts

Connect to Team Projects in Team Foundation Server

### **Other Resources**

Managing Permissions Configuring Resources to Support Team Projects Using Team Foundation Server Command-Line Tools Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Upload or Download th

Upload or Download the Microsoft Project Mapping File [TFSFieldMapping] See Also Send Feedback

Use **TFSFieldMapping** to upload and download the Microsoft Project mapping file. This file controls how Microsoft Project publishes tasks to a specified team project that was created in Visual Studio Team Foundation Server 2012. For example, you can modify the file to support fields you added to work item types or modify the way in which existing fields are published and refreshed.

You can specify which fields in Team Foundation are mapped to Microsoft Project fields. You can also control publishing behavior through the PublishOnly attribute, and designate special fields such as the context field. For the complete XML syntax of the mapping file, see The Microsoft Project Field Mapping File. This file is the same file that was defined in the process template that was used to create the team project. See Map Microsoft Project Fields to Team Foundation Fields

To access the **TFSFieldMapping** command-line tool, open a Command Prompt window where you have installed Project and either Team Explorer or TFS and enter:

## Copy Code

cd %programfiles%\Common Files\microsoft shared\Team Foundation Serv

## ⊠Note

Even if you are logged on with administrative permissions, you must open an elevated Command Prompt to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt, choose Start, open the context menu for Command Prompt, and then choose Run as Administrator. For more information, see the <u>Microsoft Web site</u>.

**Required Permissions** 

To use the **TFSFieldMapping** command, you must be a member of the Team Foundation Administrators or the Project Administrators security group. For more information, see <u>Team Foundation Server Permissions</u>.

TFSFieldMapping upload | download /collection:CollectionURL /teampro

# **Parameters**

Argument	Description
upload   download	Specifies whether to download or upload the field ma
	Specifies the URI of the team project collection in th following format: http://ServerName:Port/VirtualDirectoryName/Colle
/ <b>collection</b> :CollectionURL	If you do not specify a virtual directory is used, you the following format:
	http://ServerName:Port/CollectionName
	If you have previously connected to the server, you c the friendly name for the server instead of the URI.
/ <b>teamproject:</b> ProjectName	Specfies the name of the team project whose mappin parameters you want to modify. If the name contains enclose it in quotation marks.

/**mappingfile:**MappingFile Specifies the name and file path of the mapping file.

## To change how work item fields are mapped to Project

1. Open a Command Prompt window where you have installed Project and either Team Explorer or TFS and enter:

## Copy Code

cd %programfiles%\Common Files\microsoft shared\Team Foundation

ĭTip

Project isn't required to download the mapping file, however, it is required to upload it.

2. Download the mapping file by entering the following command:

## Copy Code

TFSFieldMapping download /collection:CollectionURL /teamproject

### ĭTip

If a name contains spaces, enclose the name in quotes.

- 3. Open the mapping file in a text editor or XML editor.
- 4. Add new mappings or edit existing mappings in the following format:

## Copy Code

```
<Mapping WorkItemTrackingFieldReferenceName="System.Id"
ProjectField=""
ProjectName=""
ProjectUnits=""
PublishOnly=""
IfSummaryRefreshOnly=""/>
```

#### -

For example, to add additional scheduling fields to a team project created using the Scrum process template, add the following mappings:

<Mapping WorkItemTrackingFieldReferenceName="Microsoft.VSTS.Sch <Mapping WorkItemTrackingFieldReferenceName="Microsoft.VSTS.Sch <Mapping WorkItemTrackingFieldReferenceName="Microsoft.VSTS.Sch <Mapping WorkItemTrackingFieldReferenceName="Microsoft.VSTS.Sch

The default Scrum template only maps the Remaining Work field to Project.

#### ⊠Note

For a full list of all the Office Project field values for the ProjectField attribute, see Field Mappings in Microsoft Project.

- 5. (Optional) Indicate if the field is to be published but not refreshed. Add the PublishOnly attribute to the mapping section for that field and set the attribute to true. This attribute can be used to allow for team members to see a field value but not be able to change it in Team Foundation.
- 6. (Optional) Indicate how summary tasks are to be refreshed in Project. Add fSummaryRefreshOnly="true" attribute to indicate that the field is never published to the work item database but is refreshed from the work item database when the row for the field is a summary task in Project, the summary task has Publish and Refresh=Yes, and the summary task contains at least one child task that is published to TFS.
- 7. Save and upload the file:

## Copy Code

TFSFieldMapping upload /collection:CollectionURL /teamproject:P

## See Also

## Reference

Field Mappings in Microsoft Project

## Concepts

The Microsoft Project Field Mapping File

## **Other Resources**

Quick tips and operational differences when tracking tasks using Project and TFS <u>Using Team Foundation Server Command-Line Tools</u>

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Specify the Bug Type to File By Using Microsoft Test Manager
See Also Send Feedback

You can enable the automatic creation of a work item when a tester finds a code defect when using Microsoft Test Manager. The **tcm** 

**bugfieldmapping** command-line tool supports the import and export of a mapping file to the team project. The mapping file defines the type of work item to create and the three data fields to be filled in by Microsoft Test Manager. The three fields are reproducible steps, system information, and the build in which the defect was found. When a tester runs a test and finds a defect, they can create a bug in which the three fields are automatically filled in.

### ⊠Note

If your team project was created with one of the default process templates provided with Team Foundation Server, the bug work item type is already enabled. However, if you are adding another type of work item to the bug category or you are working with a customized process template, you may need to use the **tcm** command.

To access the **tcm** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

## Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

For more information about additional **tcm** command-line tool options, see tcm:

Importing and Running Automated Tests for a Test Plan from the Command Line.

For more information about the bug type of work item, see Bug (Scrum), Bug (Agile), or Bug (CMMI).

**Required Permissions** 

For the team project where the work item types are defined, you must be a member of the Team Foundation Administrators security group or the Project Administrators security group. For more information, see <u>Team Foundation</u> <u>Server Permissions</u>.

#### ĭNote

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, click Start, right-click Command Prompt, and then click Run as Administrator. For more information, see the following page on the Microsoft Web site: <u>User Access Control</u>.

tcm bugfieldmapping /export /mappingfile:path /collection:Collectior

tcm bugfieldmapping /import /mappingfile:path /collection:Collectior
# **Parameters**

Parameter	Description
/export	Specifies export of the bug field mapping file. This f the work item type to create when a code defect is fo using Microsoft Test Manager. It also specifies the w fields that will be populated with data that is defined Microsoft Test Manager, such as the repro steps, syst information, and the build in which the code defect v For more information, see <u>Contents of the Bug Field</u> <u>File</u> later in this topic.
/import	Specifies import of the bug field mapping file.
	The path and file name of the XML definition file the the bug field mappings.
	⊠Note
/ <b>mappingfile</b> :Path	If you use Windows Vista, you might not have perm certain folders. If you try to export the work item tyj location where you do not have permissions, the reg virtualization technology automatically redirects the file and saves it to the virtual store. To avoid this rec you can export the file to a location where you have permissions. For more information about registry virtualization, see the following pages on the Micros site: Registry Virtualization and Common file and re virtualization issues in Windows Vista.

	/ <b>collection</b> :CollectionURL	Specifies the uniform resource identifier (URI) of the project collection. The format for the URI is as follo <b>http://ServerName:Port/VirtualDirectoryName/Coll</b>
		If no virtual directory is used, the format for the URI follows:
		http://ServerName:Port/CollectionName
	/ <b>teamproject:</b> Project	Specifies the name of the team project for which the mappings are to be applied. This team project must t in the team project collection that is specified by the parameter.
	/ <b>login:</b> UserName,Password	Optional. Specifies the name and password of a user logged on to the application-tier server for Team Fou and who has permissions to run the command. You would use this option when your Windows cred not have the appropriate permissions, you are using l
		authentication, or you are not on a domain.
	/ <b>?</b> or <b>help</b>	Displays help about the command in the Command I window.

# **Remarks**

The bug field mappings file provides Microsoft Test Manager with the name of the work item type to create and the fields to fill in for the work items that it creates.

You can define only one type of work item to be created when you run tests that use Microsoft Test Manager. Each time that you run the **tcm bugfieldmapping import** command, the work item type that is specified by the **BugFilerMappings** element replaces whatever type of work item was previously specified. For example, the following code indicates creation of the work items that are based on the type named AWBug.

### Copy Code

<BugFilerMappings workitemtypetocreate="AWBug">

For more information, see How to: Submit a Bug Using Microsoft Test Manager.

# **Contents of the Bug Field Mappings File**

The following XML syntax lists the default contents of the bug field mappings file. The file specifies the type of work item to create and the three work item fields that Microsoft Test Manager will automatically fill in, based on data that was found during the test run. All three fields must be specified when you import a bug field mappings file.

### Copy Code

```
<?xml version="1.0" encoding="utf-16"?>
<BugFilerMappings workitemtypetocreate="Bug">
<ReproSteps>Microsoft.VSTS.TCM.ReproSteps</ReproSteps>
<SystemInformation>Microsoft.VSTS.TCM.SystemInfo</SystemInformati
<BuildFoundIn>Microsoft.VSTS.Build.FoundIn</BuildFoundIn>
</BugFilerMappings>
```

For more information about fields that are used to track information that

Microsoft Test Manager finds, see Build and Test Integration Field Reference.

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URI for the team project collection: http://AdventureWorksServer:8080/tfs/Collection1
- Project name: AdventureWorks
- Bug field mapping file name: bugfieldmappings.xml
- Folder location: "C:\Users\AdminUser\Documents\"

# **Import the Bug Field Mappings File**

The following command imports the contents of the bugfieldmappings.xml file to the AdventureWorks team project that is defined in Collection1 on the server that is named AdventureWorksServer.

tcm bugfieldmapping /import /mappingfile:"C:\Users\AdminUser\Documer

# See Also

### Tasks

How to: Submit a Bug Using Microsoft Test Manager

### Concepts

Customize Process Templates How to: Run Automated Tests from the Command Line Using Tcm

#### **Other Resources**

**Using Team Foundation Server Command-Line Tools** 

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
witAdmin: Customize and Manage Objects for Tracking Work Items
See Also Send Feedback

You can change how you track your team's progress by creating and customizing objects that track work items. By using the **witadmin** command-line tool, you can create, delete, import, and export objects such as categories, global lists, global workflow, types of links, and types of work items. You can also permanently delete types of work item types, and you can delete, list, or change the attributes of fields in work item.

As the following illustration shows, you manage categories and types of and work items for each team project. You manage global lists, types of links, and fields in work item for each team project collection. You can customize global workflow for a team project or a collection.

Team Project Collection			
Global Workflow	Global Lists	Link Types	Work Item Fields
Team Project			
	0		D

#### ⊠Note

You can create and modify objects for tracking work items by using Process Editor, a power tool for Visual Studio. This tool is not supported. For more information, see the following page on the Microsoft website:

Team Foundation Server PowerTools.

In this topic

- <u>Requirements</u>
- <u>Global Parameters</u>
- <u>Command Index</u>

# Requirements

• To use the **witadmin** command-line tool, you must connect to the application-tier server for Team Foundation Server.

To access the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

# Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

A Command Prompt window opens to the directory that contains the **witadmin** command-line tool.

# **Global Parameters**

You can use the following parameters to display help text for **witadmin**.

Parameter	Description
/ <b>help</b> or / <b>?</b>	Displays the syntax and parameters for the <b>witadmin</b> command-line tool.
<b>command</b> / <b>help</b> or <b>command</b> /?	Displays the syntax and parameters for the <b>witadmin</b> command that you specify.

# **Command Index**

Tasks

Related content

Create and modify work item types. Work item types define the fields and behavior that you use to create and manage work items. You can create and modify types of work items to support the process objectives of your team and your team project.

You can manage work item types for a team project by using the following **witadmin** commands:

- **destroywitd**: Deletes a work item type.
- **exportwitd**: Exports the XML definition of a work item type.
- **importwitd**: Imports the XML definition of a work item type.
- **listwitd**: Lists the names of the work item types.
- **renamewitd**: Changes

the display name of a work item type.

Customize and Manage Work Item Types [witadmin]

• Customize Project Tracking Data, Forms, Workflow, and Other Objects

Customize process configuration. You can customize several process configuration elements to meet your Agile planning and Scrum processes. Many of these elements control the interactive tools and visual displays provided with Visual Studio Team Foundation Server 2012and Team Web Access.

- **exportagileprocessconfig**: Exports the XML definition for the process configuration (AgileConfiguration) file that customizes the backlog pages for Team Web Access.
- **exportcommonprocessconfig**: Exports the XML definition for the process configuration (CommonConfiguration) file that customizes the common areas for Team Web Access and other features.
- **importagileprocessconfig**: Imports the XML definition for the AgileConfiguration file.
- **importcommonprocessconfig**: Imports the XML definition for the CommonConfiguration file.
- Manage Process Configuration [witadmin]
- Customize the Backlog and Board Pages Using Process Configuration
- Process Configuration XML Element Reference
- Customize the Backlog Pages

Define and manage groups of work item types. By using categories, you can group work item types and then perform operations on them as a set. For example, you can specify a category when you define the criteria of a work item query to find all work items in that category, regardless of whether the items are different types. You can manage work item categories for a team project by using the following **witadmin** commands:

- **exportcategories**: Exports the XML definitions of all categories.
- **importcategories**: Imports the XML definitions of all categories.
- <u>Customize the Categories for Work Item Types [witadmin]</u>
- Define Categories to Group Work Item Types

Create, modify, export, import, and display global lists for work item types. By using global lists, you can share list items among multiple types of work items.

You can manage global lists for a collection by using the following **witadmin** commands:

- **destroygloballist**: Deletes a global list.
- **exportgloballist**: Exports the XML definition file that contains one or more global lists.
- **importgloballist**: Imports the XML file that defines one or more global lists.
- **listgloballist**: Lists the name of each global list that is defined for the collection.
- Manage Global Lists for Work Item Types [witadmin]
- Define Global Lists

Create, modify, export, or import a global workflow. By using global workflow, you can minimize the work required to define and update global lists and fields in work items that many team projects and types of work items share. You can share definitions of fields and list items among multiple types of work items.

You manage global workflow for each team project or collection by using the following **witadmin** commands:

• **exportglobalworkflow**: Exports the XML file that define a global

workflow.

- **importglobalworkflow**: Imports the XML file that defines a global workflow.
- Manage Global Workflow [witadmin]
- Customize Global Workflow

Create, modify, list, activate, deactivate, import, and export types of links. By using link types, you can associate work items based on a set of link rules. If you create custom types of links, you can track work items and their dependencies in more ways.

You can manage types of links for a collection by using the following **witadmin** commands:

- **deactivatelinktype**: Deactivates a specified type of link.
- **deletelinktype**: Permanently removes the specified type of link from the database.
- **exportlinktype**: Exports the XML definition files of one or more types of links.
- **importlinktype**: Imports the XML definition files of one or more types of links.
- **listlinktypes**: Lists the set of link types on a server.
- **reactivatelinktype**: Reactivates the type of link that you specify. You can also rename the type of link.
- <u>Customize and Manage Link Types [witadmin]</u>
- Define a Custom Link Type
- Choose Link Types to Effectively Track Your Project

Delete, list, and rename fields in types of work items, and redefine the fields that determine indexing and reporting characteristics. You use fields to find work

items and to track information in them.

You can manage fields for a collection by using the following **witadmin** commands:

- **deletefield**: Deletes the field that you specify.
- **listfields**: Lists the attributes for a field that you specify or for all fields.
- **indexfield**: Turns indexing on or off for a field that you specify.
- **changefield**: Changes one or more attributes of a field that you specify.
- Manage Work Item Fields [witadmin]
- Define Work Item Fields
- Add and Modify Work Item Fields to Support Reporting

Destroy work items. You can permanently remove work items from a team project collection by using the **witadmin destroywi** command.

• <u>Remove Work Items [witadmin]</u>

Force a rebuild of the client cache. After certain maintenance operations, you can force a rebuild of the cache of all client computers the next time that they connect to a team project collection. For example, you might force a rebuild after you move, rename, restore, or fail over a data-tier or application-tier server. By forcing a rebuild, you help prevent workspace errors during version control or build operations. To perform this operation, you use the **witadmin rebuildcache** command.

• <u>Rebuild the Client Cache [witadmin]</u>

# See Also

### Concepts

Customize Project Tracking Data, Forms, Workflow, and Other Objects

#### **Other Resources**

Determine your process and tracking customization requirements Using Team Foundation Server Command-Line Tools Send <u>feedback</u> on this topic to Microsoft. Visual Basic C#
Visual C++
F#
HLSL
JScript
Customize and Manage Work Item Types [witadmin]
See Also Send Feedback

You can manage work item types for a team project by using the following **witadmin** commands:

- **destroywitd**: Destroys a work item type, and destroys every work item of that type permanently without recovery.
- **exportwitd**: Exports the definition of a work item type to an XML file, or to the Command Prompt window.
- **importwitd**: Imports work item types from an XML definition file into a team project on a server that runs Team Foundation Server. If a work item type with the same name already exists, the new work item type definition overwrites the existing one. If the work item type does not already exist, this command creates a new work item type. To validate the XML that defines a work item type, but not import the file, you use the /**v** option.
- **listwitd**: Displays the names of the work item types in the specified project in the Command Prompt window.
- **renamewitd**: Changes the display name of a work item type within a specific project. After you run this command, work items of this type show the new name.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

### Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

For more information about work item types, see Add Type Definitions for Work Items to a Process Template.

### ⊠Note

You can create and modify work item types by using Process Editor, a power tool for Visual Studio. This tool is not supported. For more information, see the following page on the Microsoft Web site: <u>Team Foundation Server Power</u> <u>Tools</u>.

### Requirements

For the team project where the work item types are defined, you must have the following permissions set:

- To export or list work item types, you must be a member of the Project Administrators group or have your View project-level information permission set to Allow.
- To destroy, import, or rename work item types, you must be a member of the Team Foundation Administrators security group or the Project Administrators security group.

For more information, see <u>Team Foundation Server Permissions</u> and <u>Team</u> <u>Foundation Server Default Groups</u>, <u>Permissions</u>, and <u>Roles</u>.

#### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the shortcut menu for the Command Prompt, and then choose Run as Administrator. For more information, see the Microsoft Web site: <u>User Access Control</u>.

witadmin destroywitd /collection:CollectionURL /p:Project /n:TypeNam witadmin exportwitd /collection:CollectionURL /p:Project /n:TypeNam@ witadmin importwitd /collection:CollectionURL [/p:Project] /f:FileN@ witadmin listwitd /collection:CollectionURL /p:Project

witadmin renamewitd /collection:CollectionURL /p:Project /n:TypeName

# **Parameters**

Parameter	Description
/ <b>collection</b> :CollectionURL	Specifies the URI of the team project collection. The the URI is the following: http://ServerName:Port/VirtualDirectoryName/Colle If no virtual directory is used, then the format for the following: http://ServerName:Port/CollectionName
/ <b>p:</b> Project	The team project for which the types of work items a managed. This team project must be defined in the pr collection specified by the / <b>collection</b> parameter. The / <b>p</b> parameter is required unless you run the <b>impo</b> command with the / <b>v</b> option.
/ <b>n:</b> TypeName	The name of the work item type to destroy, export, in rename.
	The path and file name of the XML definition file tha the types of work items to be exported or imported. If this parameter when you use the <b>exportwitd</b> commar XML appears in the Command Prompt window.
	⊠Note
/ <b>f:</b> FileName	If you are using Windows Vista you might not have permissions to certain folders. If you try to export the

	item type to a location where you do not have permis registry virtualization technology automatically redir exported file and saves it to the virtual store. To avoid redirection, you can export the file to a location when have permissions. For more information about regists virtualization, see the Microsoft Web site: <u>Registry</u> <u>Virtualization</u> and <u>Common file and registry virtualization</u> issues in Windows Vista.
/ <b>e:</b> Encoding	The name of a .NET Framework 2.0 encoding format command uses the specified encoding to export or im XML data. For example, /e:utf-7 specifies Unicode (I encoding. If you omit this parameter, <b>witadmin</b> tries the encoding, and if detection fails, <b>witadmin</b> uses U
/exportgloballists	Exports the definitions of global lists referenced by the item type. The definitions for global lists will be embed the work item type definition XML. When not specific definitions for global lists are omitted.
	Validates the XML that defines the work item type, bi import the XML definition file.
/ <b>v</b>	⊠Note
	You can validate the type definition without specifyin project. References to project-scoped groups is ignor
/ <b>new:</b> NewName	The new name of the work item type.
/noprompt	Disables the prompt for confirmation.
	Displays help about the command in the Command P

/? or help

window.

# **Remarks**

When you use the **destroywitd** command, it destroys all the following objects:

- The work item type
- All work items of that type
- Corresponding entries in the work item tables, the long text tables, and the link tables
- Objects in the work item type metadata cache

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URI for the team project collection: http://AdventureWorksServer:8080/tfs/DefaultCollection
- Project name: AdventureWorks
- Input or output file name: myworkitems.xml
- Work item type name: myworkitem
- Default encoding: UTF-8

# **Export the Definition of a Work Item Type**

The following command exports the definition for myworkitem to the file, myworkitems.xml.

witadmin exportwitd /collection:http://AdventureWorksServer:8080/tfs

The following example exports the work item by using Unicode (UTF-7) encoding.

witadmin exportwitd /collection:http://AdventureWorksServer:8080/tfs

# ■Export the Definition of a Work Item Type and Its Referenced Global Lists

The following example exports the work item type and its referenced global lists.

witadmin exportwitd /collection:http://AdventureWorksServer:8080/tfs

# **List the Definition of a Work Item Type**

The following example displays the definition of the work item type the Command Prompt window.

#### Copy Cod

witadmin exportwitd /collection:http://AdventureWorksServer:8080/tfs

# **Import the Definition of Work Item Types**

The following example imports the work item definition from the XML file.

#### 

witadmin importwitd /collection:http://AdventureWorksServer:8080/tfs

### **Solution State Series and Series**

The following example validates the XML that defines the work item type but does not import the definition.

#### 

witadmin importwitd /collection:http://AdventureWorksServer:8080/tfs

# See Also

### Tasks

Create a Work Item Type

# Concepts

Customize Project Tracking Data, Forms, Workflow, and Other Objects

### **Other Resources**

witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Customize the Categories for Work Item Types [witadmin]
See Also Send Feedback

You can manage the categories for work item types defined for a team project by using the following **witadmin** commands:

- **exportcategories**: Exports the XML definition of categories defined on a server that runs Team Foundation Server.
- **importcategories**: Imports a work item type XML definition file into a team project on a server that runs Team Foundation Server.

If a category with the same name already exists, this command overwrites the existing definition. If the work item type does not already exist, this command creates a new category.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

### Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace %programfiles% with %programfiles(x86)%.

#### ⊠Note

You can create and modify categories by using Process Editor, a power tool for Visual Studio. This power tool is not supported. For more information, see the following page on the Microsoft Web site: <u>Team Foundation Server Power</u> <u>Tools.</u>

#### Requirements

For the team project for which the categories are defined, you must have the following permissions set:

- To export categories of work item types, you must have your View projectlevel information permission set to Allow.
- To import categories of work item types, you must be a member of the Project Administrators security group or have the Edit project-level information permission set to Allow.

For more information, see <u>Team Foundation Server Permissions</u> and <u>Team</u> <u>Foundation Server Default Groups</u>, <u>Permissions</u>, and <u>Roles</u>.

#### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the Command Prompt shortcut menu, and choose Run as Administrator. For more information, see the Microsoft Web site: <u>User Account Control</u>.

witadmin exportcategories /collection:CollectionURL /p:Project [/f:F

witadmin importcategories /collection:CollectionURL /p:Project /f:Fi

# **Parameters**

	Parameter	Description
		Specifies the URI of the team project collection. The the URI is the following: http://ServerName:Port/VirtualDirectoryName/Colle
/ <b>collection</b> :CollectionUR	/collection:CollectionURL	If no virtual directory is used, then the format for the following:
		http://ServerName:Port/CollectionName.
	/ <b>p</b> :Project	The name of the team project from which the categor exported or to which the categories are imported.
	/ <b>f</b> :FileName	The path and file name of the XML definition file tha the categories to be exported or imported. If you omit parameter when you use the <b>exportcategories</b> comm command lists the categories in the Command Promp
	/ <b>e</b> :Encoding	The name of a .NET Framework 2.0 encoding format specified encoding will be used to export or import th data. For example, /e:utf-7 specifies Unicode (UTF-7 encoding. If you omit this parameter, <b>witadmin</b> atten detect the encoding, and if detection fails, <b>witadmin</b> 8.
	/ <b>?</b> or <b>help</b>	Displays help about the command in the Command P window.

# **Remarks**

Importing a categories file creates the categories if they do not already exist. If categories already exist, the **witadmin importcategories** command will warn you that the current list will be overwritten. You can write your own program to update an existing set of categories, or you can update the categories yourself with new data.

#### **▲Important**

Changes you make to categories can impact process configuration. See Customize the Backlog and Board Pages Using Process Configuration.

For information about the categories XML file structure, see Categories Definition Schema Reference.

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URI for the team project collection: http://AdventureWorksServer:8080/tfs/DefaultCollection
- Project name: AdventureWorks
- Input or output file name: myCategories.xml
- Default encoding: UTF-8

# **List Work Item Type Categories**

The following command lists the work item type categories defined for the AdventureWorks project.

### Copy Code

witadmin exportcategories /collection:http://AdventureWorksServer:80

# **Export Work Item Type Categories**

The following command exports the work item type categories defined for the AdventureWorks project to the myCategories XML file.

#### **₽C**o

witadmin exportcategories /collection:http://AdventureWorksServer:80

# **Import Work Item Type Categories**

The following example imports categories from the myCategories XML file to the AdventureWorks project.

#### **₽C**0

witadmin importcategories /collection:http://AdventureWorksServer:80

# See Also

### Concepts

Customize the Backlog and Board Pages Using Process Configuration

#### **Other Resources**

Define Categories to Group Work Item Types Work Item Type Categories Element Reference witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Manage Global Lists for Work Item Types [witadmin]
See Also Send Feedback

Global lists are pick lists that you can include in one or more fields and types of work items. Use them to minimize the work that is required to update a list that is shared by multiple types of work items. Because global lists are defined for a team project collection, they can be included in work item types for all team projects within the collection.

#### ⊠Note

Global lists must not include project-scoped security groups because global lists do not have the same scope as a team project.

You can define a global list within its own definition file, in the definition file for a type of work item, or in global workflow. The global-list definition manages the global lists that are defined for a collection. The global-list definition uses the following commands in the **witadmin** command-line tool:

- **destroygloballist**: Destroys a global list.
- **exportgloballist**: Exports the global lists to an XML file, or to the Command Prompt window. It exports a single file, which contains all global lists that have been defined for the team project collection.
- **importgloballist**: Imports global lists from an XML file. If you try to import a global list that already exists on the server, a warning asks if you want to overwrite the global list that is on the server. If you try to import a global list that does not already exist, the command creates a new global list.

• **listgloballist**: Displays the name of each global list defined on the server.

See Define Global Lists and Customize Global Workflow.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

# Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

#### ⊠Note

You can create and modify global lists by using Process Editor, a power tool for Visual Studio. This tool is not supported. For more information, see the following page on the Microsoft Web site: <u>Team Foundation Server Power</u> <u>Tools</u>.

### Requirements

For the team project collection where the global lists are defined, you must have the following permissions set:

- To export or list global lists, you must be a member of the Project Collection Valid Users group or have your View collection-level information permission set to Allow
- To destroy or import global lists, you must be a member of the Team Foundation Administrators security group or the Project Administrators security group.

See <u>Team Foundation Server Permissions</u> and <u>Team Foundation Server Default</u> <u>Groups, Permissions, and Roles</u>.

⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the Command Prompt shortcut window, and choose Run as Administrator. For more information, see this page on the Microsoft Web site: <u>User Access Control</u>.

witadmin destroygloballist /collection:CollectionURL /n:GlobalListNa

witadmin exportgloballist /collection:CollectionURL [/f:FileName] [/

witadmin importgloballist /collection:CollectionURL /f:FileName [/e:

witadmin listgloballist /collection:CollectionURL
# **Parameters**

	Parameter	Description
	/ <b>collection</b> :CollectionURL	Specifies the URI of the team project collection. The the URI is the following: <b>http</b> ://ServerName:Port/VirtualDirectoryName/Colle
		If no virtual directory is used, then the format for the following:
		http://ServerName:Port/CollectionName.
	/ <b>n</b> :GlobalListName	The name of the global list to destroy.
	<b>/f</b> :FileName	The path and the name of the global list XML definiti export or import.
		⊠Note
		If the client computer runs Windows Vista, you migh permissions to certain folders. If you try to export the list to a location where you do not have permissions, registry virtualization technology automatically redir exported file and saves it to the virtual store. For mor information, see the following pages on the Microsof site: Registry Virtualization and Common file and rej virtualization issues in Windows Vista. To avoid this redirection, you can export the file to a location wher have permissions.
		The name of a .NET Framework 2.0 encoding format

/ <b>e:</b> Encoding	specified encoding will be used to export or import th data. For example, /e utf-7 specifies Unicode (UTF-7 encoding. If you omit this parameter, <b>witadmin</b> atten detect the encoding, and if detection fails, <b>witadmin</b> 8.
/noprompt	Disables the prompt for confirmation.
/ <b>?</b> or <b>help</b>	Displays help about the command in the Command P window.

# **Remarks**

Importing a global list creates a list if one does not exist. If the list already exists, the **witadmin importgloballist** command will warn you that the current list will be overwritten. You can write your own program to update an existing global list, or you can update the global list yourself with the new data.

To create a new global list, start with the following code and modify it as needed. This example defines a global list of disciplines that you can assign to tasks.

Copy Code

```
<?xml version="1.0" encoding="utf-8"?>
<gl:GLOBALLISTS xmlns:gl="http://schemas.microsoft.com/VisualStudio/
<GLOBALLIST name="Disciplines">
<LISTITEM value="Architecture" />
<LISTITEM value="Requirements" />
<LISTITEM value="Development" />
<LISTITEM value="Project Management" />
<LISTITEM value="Project Management" />
<LISTITEM value="Test" />
</GLOBALLIST>
</gl:GLOBALLIST>
```

To add a global list to a field, export the definition for the work item type that contains the field and add it to the field definition, as shown in the following example:

#### Copy Code

To view the changes, import the type definition and refresh your browser or

client cache. You might need to close any work items that reference the field and reopen them.

For information about export and import of type definitions, see <u>Customize and</u> <u>Manage Work Item Types [witadmin]</u>.

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URI for the team project collection: http://AdventureWorksServer:8080/tfs/DefaultCollection
- Server Web site port number: 8080

### **Display the names of global lists**

The following example shows the global lists defined on the server. The example exports the global lists using the default UTF-8 encoding:

### Copy Code

witadmin listgloballist /collection:http://AdventureWorksServer:808@

### **Export the definition of global lists**

The following example exports the global lists:

### Copy Code

witadmin exportgloballist /collection:http://AdventureWorksServer:80

The following example exports the same global lists to the same server, but uses Unicode (UTF-7) encoding:

#### 🖹 Сору (

witadmin exportgloballist /collection:http://AdventureWorksServer:80

**Import the definition of global lists** 

The following example imports global lists:

Copy Code

witadmin importgloballist /collection:http://AdventureWorksServer:80

The following example imports the same global lists to the same server, but uses Unicode (UTF-7) encoding:

Copy (

witadmin importgloballist /collection:http://AdventureWorksServer:80

# See Also

### Concepts

Define Global Lists

### **Other Resources**

Customize Global Workflow witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Manage Global Workflow [witadmin]
See Also Send Feedback

By using global workflow, you can minimize the work required to define and update work item fields and global lists that multiple team projects and types of work items share. With global workflow, you can define and update fields and global lists. You can manage the global workflow for a team project collection or a team project by using the following commands in the **witadmin** command-line tool:

- **exportglobalworkflow**: Exports the global workflow to an XML file or the Command Prompt window.
- **importglobalworkflow**: Imports global workflow from an XML file.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

### Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

For more information about global workflow, see Customize Global Workflow.

Requirements

• To export a global workflow, you must be a valid user of the team project or team project collection.

- To import a global workflow to a team project, you must be a member of the following security groups: Team Foundation Administrators, Project Collection Administrators, or Project Administrators.
- To import a global workflow to a team project collection, you must be a member of either the Team Foundation Administrators or Project Collection Administrators.

See <u>Team Foundation Server Permissions</u> and <u>Team Foundation Server Default</u> <u>Groups, Permissions, and Roles</u>.

#### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the Command Prompt shortcut menu, and then choose Run as Administrator. For more information, see the following page on the Microsoft website: <u>User Access Control</u>.

witadmin exportglobalworkflow /collection:CollectionURL [/p:Project]

witadmin importglobalworkflow /collection:CollectionURL [/p:Project]

# **Parameters**

	Parameter	Description
	/ <b>collection</b> :CollectionURL	Specifies the URI of the collection. You must specify in the following format: http://ServerName:Port/VirtualDirectoryName/Colle
		If you do not specify a virtual directory, you must spe URI in the following format:
		http://ServerName:Port/CollectionName.
	/ <b>p:</b> ProjectName	Optional. The team project for which you want to exp import the global workflow. This team project must b in the collection that you specified by using the /colle parameter. If you do not specify a team project, the gl workflow is imported or exported for the collection.
		The path and the name of the XML definition file for workflow to export or import.
		<b>⊠Note</b>
	/ <b>f</b> :FileName	If the client computer is running Windows Vista, you not have permissions to certain folders. If you try to ( global list to a location where you do not have permi- registry virtualization technology automatically redir exported file and saves it to the virtual store. For mor information, see the following pages on the Microsof site: <u>Registry Virtualization</u> and <u>Common file and rej</u> <u>virtualization issues in Windows Vista</u> . To avoid this redirection, you can export the file to a location wher

	have permissions.
/ <b>e:</b> Encoding	Optional. The name of a .NET Framework 2.0 encodi The specified encoding will be used to export or impo XML data. For example, /e utf-7 specifies Unicode (I encoding. If you omit this parameter, <b>witadmin</b> atten detect the encoding and uses UTF-8 if detection fails.
/v	Optional. Validates the XML that defines the global v but does not import the definition file.
/exportgloballists	Optional. Exports the definitions of global lists that the workflow references. The definitions for global lists were embedded into the XML definition of the global work you do not specify this parameter, the definitions for are omitted.
/ <b>?</b> or <b>help</b>	Displays help about the command in the Command P window.

# **Remarks**

You can define work item fields by importing them through a global workflow. However, you cannot change the properties of existing fields by using global workflow. If you import a global workflow that does not contain a **FIELDS** element, all previously imported rules for global workflow will be deleted. Field definitions will not be affected.

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URI for the collection: http://AdventureWorksServer:8080/tfs/DefaultCollection
- Team project: Contoso
- Port number for the server website: 8080

## **Export the definition of a global workflow for a team project**

The following example exports the global workflow for a team project:

witadmin exportglobalworkflow /collection:http://AdventureWorksServe

### **Import the definition of a global workflow to a team project**

The following example imports the global workflow to the Contoso team project:

witadmin importglobalworkflow /collection:http://AdventureWorksServe

### **Import the definition of a global workflow to a collection**

The following example imports the global workflow to the collection:

**₽C**0

witadmin importglobalworkflow /collection:http://AdventureWorksServe

# See Also

### Concepts

Define Global Lists

### **Other Resources**

Customize Global Workflow witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Manage Process Configuration [witadmin]
See Also Send Feedback

By customizing the process configuration, you can customize the layout and features available through the backlog and task board pages of Team Web Access. You can configure the work items that are listed, the order in which they are listed, and the fields that appear and are used to calculate burndown and capacity.

#### ⊠Note

If you have customized your process template or team project, you may need to customize the process configuration. For more information, see Customize the Backlog and Board Pages Using Process Configuration.

You can manage the process configuration for a team project by using the following commands in the **witadmin** command-line tool:

- **exportagileprocessconfig**: Exports the process configuration definition that customizes the backlog pages to an XML file or the Command Prompt window.
- **exportcommonprocessconfig:** Exports the process configuration definition that is used in common to an XML file or the Command Prompt window.
- **importagileprocessconfig**: Imports the process configuration definition that customizes the backlog pages from an XML file.
- **importcommonprocessconfig**: Imports the process configuration definition that is used in common from an XML file.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

### Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

# On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

Requirements

- To export process configuration definitions, you must be a valid user of the team project or collection.
- To import process configuration definitions, you must be a member of the following security groups: Team Foundation Administrators or Project Administrators.
- To verify process configuration and access the Agile planning tools available only from Team Web Access, your Team Foundation Server administrator must grant you Web Access Permissions. Additional licensing requirements may apply. For more information, see <u>Change access levels</u>.

For more information, see <u>Team Foundation Server Permissions</u> and <u>Team</u> <u>Foundation Server Default Groups</u>, <u>Permissions</u>, <u>and Roles</u>.

### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to run the witadmin command on a server that is running Windows Server 2008. To open an elevated Command Prompt window, click Start, right-click Command Prompt, and then click Run as Administrator. For more information, see the following page on the Microsoft website: <u>User Access Control</u>.

witadmin exportagileprocessconfig /collection:CollectionURL /p:Proj@

witadmin exportcommonprocessconfig /collection:CollectionURL /p:Proj

witadmin importagileprocessconfig /collection:CollectionURL [/p:Proj

witadmin importcommonprocessconfig /collection:CollectionURL [/p:Prc

# **Parameters**

	Parameter	Description
	/ <b>collection</b> :CollectionURL	Specifies the URL of the collection. You must specify in the following format: http://ServerName:Port/VirtualDirectoryName/Colle
		If you do not specify a virtual directory, you must spe URI in the following format:
		http://ServerName:Port/CollectionName.
	/ <b>p</b> :ProjectName	Required. The team project for which you want to exj import the process configuration. This team project n defined in the collection that you specified by using t /collection parameter. You do not need to specify a te when you specify the /v switch.
		The path and the name of the XML definition file for process configuration to export or import.
		<b>⊠Note</b>
	/ <b>f</b> :FileName	If the client computer is running Windows Vista, you have permissions to certain folders. If you try to expe global list to a location where you do not have permit registry virtualization technology automatically redir exported file and saves it to the virtual store. For mor information, see the following pages on the Microsof Registry Virtualization and Common file and registry virtualization issues in Windows Vista. To avoid this redirection, you can export the file to a location wher

		have permissions.
/	e:Encoding	Optional. The name of a .NET Framework 2.0 encodi The specified encoding will be used to export or impo XML data. For example, /e utf-7 specifies Unicode (I encoding. If you omit this parameter, <b>witadmin</b> atten detect the encoding and uses UTF-8 if detection fails.
/	v	Optional. Validates the XML that defines the process configuration but does not import the definition file.
/	<b>?</b> or <b>help</b>	Displays help about the command in the Command P window.

# **Remarks**

If you are updating an upgraded team project and importing process configuration files for the first time, you must import the definition file for CommonConfiguration first, and then import the definition file for AgileConfiguration. Also, the Categories file must contain the definitions for the categories referenced in the CommonConfiguration file. For more information, see Configure features after a TFS upgrade.

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URL for the collection: http://AdventureWorksServer:8080/tfs/DefaultCollection
- Team project: Contoso
- Port number for the server website: 8080

## **Export the Agile Configuration Definition for a Team Project**

The following example exports the process configuration for the team project backlog pages:

witadmin exportagileprocessconfig /collection:http://AdventureWorks

### **Import the Agile Configuration Definition for a Team Project**

The following example imports the process configuration for the backlog pages to the Contoso team project:

witadmin importagileprocessconfig /collection:http://AdventureWorks

# See Also

### Concepts

Configure features after a TFS upgrade Customize the Backlog and Board Pages Using Process Configuration Process Configuration XML Element Reference

### **Other Resources**

witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Customize and Manage Link Types [witadmin]
See Also Send Feedback

Similar to work item types, you can define and modify custom link types. However, you can't make any changes to system-defined link types. Before you add a new link type to your team project collection, review the link types available for your use. See Link work items and objects to support traceability.

You manage the link types defined for a team project collection by using the following **witadmin** commands:

- **deactivatelinktype**: Deactivates the specified link type. Users will no longer be able to create links of this type. Existing links of this type will continue to function correctly.
- **deletelinktype**: Permanently removes the specified link type from the database. All links defined with this link type are also removed.
- **exportlinktype**: Exports the definitions of link types. You can export the definition of a single link type or all link types defined for the server.
- **importlinktype**: Imports the definitions of link types from an XML file. If a link type with the same reference name already exists, it will be updated to match the imported link type. If the link type does not already exist, a new link type will be created.
- **listlinktypes**: Lists the available set of link types on a server.
- **reactivatelinktype**: Reactivates the specified link type, and optionally assigns it a new name.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

### Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

#### Requirements

For the team project collection where the link types are defined, you must have the following permissions set:

- To list or export link types, you must have your View project-level information permission for a team project in the collection set to Allow.
- To delete, import, or reactive link types, you must be a member of the Project Collection Administrators security group or have your Manage work item link types permission set to Allow.

For more information, see <u>Team Foundation Server Permissions</u> and <u>Team</u> <u>Foundation Server Default Groups</u>, <u>Permissions</u>, <u>and Roles</u>.

#### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the Command Prompt shortcut menu, and then choose Run as Administrator. For more information, see the Microsoft Web site: <u>User Access Control</u>.

witadmin deactivatelinktype /collection:CollectionURL /n:LinkName

witadmin deletelinktype /collection:CollectionURL /n:LinkName [/nop)

witadmin exportlinktype /collection:CollectionURL [/n:LinkName] [/f:

witadmin importlinktype /collection:CollectionURL /f:FileName [/e:Er

witadmin listlinktypes /collection:CollectionURL

witadmin reactivatelinktype /collection:CollectionURL /n:LinkName

# **Parameters**

Parameter	Description
	Specifies the URI of the team project collection. The the URI is the following: http://ServerName:Port/VirtualDirectoryName/Colle
	For example:
/ <b>collection</b> :CollectionURL	http://ServerName:8080/tfs/Collection0
	If a virtual directory is not used, then the format for tl the following:
	http://ServerName:Port/CollectionName
/ <b>n:</b> LinkName	The name or the reference name of the link type to de delete.
/ <b>f:</b> FileName	The XML file of link types. Required for import, opti export. If you omit this parameter, the command outp on the display.
/ <b>e:</b> encoding	The name of a .NET Framework 2.0 encoding format specified encoding will be used to export or import th data. For example, /e:utf-7 specifies Unicode (UTF-7 encoding. If you omit this parameter, <b>witadmin</b> atten detect the encoding, and if detection fails, <b>witadmin</b> 8.

/ <b>v</b>	Validates the link type XML without importing it.
/noprompt	Disables the prompt for confirmation.
/ <b>?</b> or <b>help</b>	Displays help about the command in the Command P window.

# **Remarks**

For the structure of the link type definition schema, see Link Type Element Reference.

For information about how you can restrict the types of link relationships that team members can make, see Link Controls, Restrictions, and Field Reference and LinksControlOptions Elements.

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URI for the team project collection: http://AdventureWorksServer:8080/tfs/DefaultCollection
- Server name: AdventureWorksServer
- Input or output file name: myLinkTypes.xml
- Link type name: mylinktype
- Default encoding: UTF-8

## **■List Link Types**

The following command displays the custom link types defined for Team Foundation AdventureWorksServer.

### Copy Code

witadmin listlinktypes /collection:http://AdventureWorksServer:8080,

## **List the Definition of a Link Type**

The following example displays the definition of the link type in the Command Prompt window:

### Copy Code

witadmin exportlinktype /collection:http://AdventureWorksServer:808@

## **Deactivate and then Reactivate a Link Type**

The following examples deactivate the link type, mylinktype, and then reactivate

### Copy Code

```
witadmin deactivatelinktype /collection:http://AdventureWorksServer:
witadmin activatelinktype /collection:http://AdventureWorksServer:80
```

### **Export the Definition of a Link Type**

The following command exports the definition of all link types to the file, mylinktype.xml:

### Copy Code

witadmin exportlinktype /collection:http://AdventureWorksServer:808@

### **Import the Definition of Link Types**

The following example imports the definition of the link types defined in the XML file:

### Copy Code

witadmin importlinktype /collection:http://AdventureWorksServer:808@

it:

## See Also

### Concepts

Link Type Element Reference Link Controls, Restrictions, and Field Reference

#### **Other Resources**

Link work items and objects to support traceability witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Manage Work Item Fields [witadmin]
See Also Send Feedback

You can manage the work item type fields that are defined for a team project collection by using the following **witadmin** commands:

- **changefield**: Changes one or more attributes of a field. When you change one of the following attributes, you change it for all work item types and team projects within the team project collection:
  - Data type for PlainText or HTML fields.

### **▲Important**

When you upgrade Team Foundation Server from an earlier version to the current version, the type assignment for the Description (System.Description) field is automatically converted from PlainText to HTML. With the changefield command, you can restore the content of this field to display plain text.

- Friendly name that displays in the work item query. This name may differ from that displayed on the work item form.
- Reporting attributes which includes the name of the field as it appears in a report, the reference report name, and the reporting type.
- Synchronization with Active Directory you can enable/disable synchronization of person name fields.
- **deletefield**: Deletes the specified field.
- **indexfield**: Turns indexing on or off for the specified field. When you

enable indexing for a field, you may increase the performance of finding work items whose queries specify that field. If you add a custom field that you use in many of your work item queries, you may want to enable indexing for that field.

• **listfields**: Lists the attributes for all fields or a specified field.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

### Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

For an overview of the fields defined within the default process templates provided with Team Foundation Server, see Work Item Field Reference for Visual Studio ALM.

### ⊠Note

You can create and modify work item fields by using Process Editor, a power tool for Visual Studio. This tool is not supported. For more information, see the following page on the Microsoft Web site: <u>Team Foundation Server Power</u> <u>Tools</u>.

### Requirements

- To list fields, you must have your View project-level information permission for the team project in the collection set to Allow.
- To delete or rename fields or change an attribute of a field, you must be a member of the Team Foundation Administrators security group or the Project Collection Administrators security group.

For more information, see <u>Team Foundation Server Permissions</u> and <u>Team</u>

Foundation Server Default Groups, Permissions, and Roles.

#### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the Command Prompt shortcut menu, and then choose Run as Administrator. For more information, see the Microsoft Web site: <u>User Access Control</u>.

witadmin changefield /collection:CollectionURL /n:RefName [/name:NewName] [/syncnamechanges:true | false] [/reportingname:ReportingName] [/reportingrefname:ReportingRefName] [/reportingtype:Type] [/reportingformula:Formula] [/type:PlainText | HTML] [/noprompt]

witadmin deletefield /collection:CollectionURL /n:RefName [/noprompt

witadmin indexfield /collection:CollectionURL /n:Name /index:on|off

witadmin listfields /collection:CollectionURL /n:RefName [/unused]

## **Parameters**

Parameter	Description
	Specifies the URI of the team project co the URI is the following: <b>http</b> ://ServerName:Port/VirtualDirectory
/ <b>collection</b> :CollectionURL	If no virtual directory is used, then the for following:
	http://ServerName:Port/CollectionName
/ <b>n</b> :RefName / <b>n</b> :Name	The reference name of a work item type
/index	Specifies to enable or disable indexing for Specify <b>on</b> to enable indexing and <b>off</b> to
/ <b>name:</b> NewName	Specifies the new name for the field.
/syncnamechanges	Specifies to use the work item field to st update as changes are made in Active D workgroup. This option is valid only wh type of String is specified for the typena Specify true to enable synchronization for false to disable synchronization for the c

/ <b>reportingname</b> :ReportingName	Specifies the name of the field in the dat for reporting.
/ <b>reportingrefname</b> :ReportingRefName	Specifies the reference name of the field to be used for reporting.
	Specifies how the field is used in the wa The following values are valid:
/ <b>reportingtype:</b> Туре	• <b>dimension:</b> Used for the Integer, Stifields.
	• <b>detail:</b> Used for the Integer, Double fields.
	• <b>measure:</b> Used for the Integer and default aggregation type is sum. Yo aggregation type by using the <b>form</b>
	• <b>disable:</b> Used to disable reportabili
	For more information, see Working with
/ <b>reportingformula:</b> Formula	Specifies the aggregation formula to be reported as a measure. The only support
/ <b>type:</b> HTML   PlainText	Specifies to convert the contents of the f HTML or from HTML to PlainText. You option only for fields whose type assign HTML. See FIELD (Definition) Elemen
/unused	Lists all fields that are not used by any to the team project collection.
/nopromptDisables prompt for confirmation./? or helpDisplays help about the command in the window.

# **Remarks**

## **∃Indexed fields**

A query index is created based on those fields that have indexing enabled. This index improves the response time when running queries that include indexed fields.

By default, the following fields are indexed: Assigned To, Created Date, Changed By, State, Reason, Area ID, Iteration ID, and Work Item Type. If there are other fields that your team frequently uses in their queries, you can add them to the query index.

## **Synchronizing person names with Active Directory**

You must manually enable synchronization of any custom work item fields that are used to assign person names that reference Active Directory. You must enable synchronization for each field for each team project collection that contains the custom fields.

All system reference fields that show person-names have the attribute syncnamechanges set to true. Such fields include System.AuthorizedAs, System.AssignedTo, System.ChangedBy and System.CreatedBy. Synchronization in enabled for each person name field that is defined in one of the default process templates. For more information, see Assignments and Workflow Field Reference.

After synchronization is enabled, the field no longer shows a static string. Instead, the field shows the name associated with a user account. When you change the user name in Active Directory or in Workgroup, a field with syncnamechanges set to true automatically shows the new name.

When you assign the syncnamechanges attribute to a String field, the field always accepts valid user names. However, the field does not allow group names that are stored in Team Foundation Server or in Active Directory if any one of the following conditions is true:

- The VALIDUSER rule is specified across all work item types
- The VALIDUSER rule is specified for a work item type
- The ALLOWEDVALUES rule is specified for a work item type, and that rule has a filter criteria that excludes groups

For more information, see All FIELD XML Elements Reference.

## ■Attributes that you can change for each work item type

You change the following attributes or values defined for a field by changing the work item type definition in which the field appears:

- Name that displays on the work item form. See Control XML Element Reference.
- Help text. See Define the Help Text for a Work Item Field.
- Allowed values or items within a pick list or drop-down menu. See Define User Lists, Pick Lists, and Global Lists.

# **Examples**

Unless otherwise specified, the following values apply in each example:

- URI for the team project collection: http://AdventureWorksServer:8080/tfs/DefaultCollection
- Work item field name: AdventureWorks.Field
- Default encoding: UTF-8

## ∃List fields

Use **witadmin listfields** to see the set of fields in use, to select one to add to a work item type. Also, you can list the attribute assignments defined for a specific field and determine which fields are used by which team projects.

## View the attributes of a work item field

• Enter the following command to list the attributes defined for a specified field, such as Microsoft.VSTS.Common.Issue.

#### Сору

witadmin listfields /collection:http://AdventureWorksServer:808

Field and attribute information appears for the named field, as shown in this example.

```
Copy Code
```

Field: Microsoft.VSTS.Common.Issue Name: Issue Type: String Reportable As: dimension Use: Adventure Works (Shared Steps), AW Future (Shared Steps), / Indexed: False The Use parameter indicates the name of each project and the work item type where the field is used. For more information about field attributes, see Work Item Field Reference for Visual Studio ALM.

### List all fields in a team project collection

• Enter the following command to list all fields defined for a team project collection.

#### Copy Code

witadmin listfields /collection:http://AdventureWorksServer:8080

Field information for all the fields for the named project collection appears. See Work Item Field Reference for Visual Studio ALM.

#### List fields that are not being used

• Enter the following command to list the fields that are no longer being used in the team project collection by any work item type.

#### Copy Code

witadmin listfields /collection:http://AdventureWorksServer:808

Field and attribute information appears for each field that is not being used, as shown in this example.

#### Copy Code

Field: Microsoft.VSTS.CMMI.TaskType Name: Task Type Type: String Reportable As: dimension Use: Not In Use Indexed: False Field: Microsoft.VSTSUE.Common.Flag Name: Flag Type: String

```
Reportable As: dimension
Use: Not In Use
Indexed: False
Field: Microsoft.VSTSUE.Common.Progress
Name: Progress
Type: String
Reportable As: dimension
Use: Not In Use
Indexed: False
```

## **■Rename a field**

You can change the friendly name of a work item field to meet the naming conventions that your team uses. Note that the new name is applied to all work item types that reference the changed field in all team projects in the project collection. The friendly name displays when you define filter criteria in a work item query. The name that appears on a work item form may be different than the friendly name defined for the field.

### To rename a work item field

1. Enter the following command to rename the friendly name assigned to Microsoft.VSTS.Common.Rank to Important Rank.

witadmin changefield /collection:http://AdventureWorksServer:80{

2. At the confirmation prompt type **y** and then press ENTER. Wait until you receive the following message:

#### Field renamed.

#### To verify changes imported to a single project

1. In Team Explorer, choose 🖗 Refresh.

The latest updates are downloaded from the server, which include the changes that you just made to the field name. Wait several seconds for the

refresh to finish.

- 2. Choose New Query to create a query.
- 3. In the Query Editor, choose the Click here to add a clause link to add a row, select the blank Field cell, and in the cell, type Rank. The following message that appears above the results list. This message indicates that the Rank cannot be found.

Run the query to see the query results. TF51005: The query references a field that does not exist. The error is caused by <<Rank>>.

- 4. Delete the value Rank from the Field cell, and type Important Rank into the cell.
- 5. Select <> in the Operator cell, and type 1 into the Value cell.
- 6. On the Query toolbar, choose ▶ Run.
- 7. Open the shortcut menu for any row in the results and select Column Options. Scroll down in the Available columns list. Notice that the Rank field is no longer present but the Important Rank field is present.
- 8. Select Important Rank in the Available columns box, and then choose the > button (add selected columns). Choose OK.

Notice that the friendly name for Microsoft.VSTS.Common.Rank has been renamed from Rank to Important Rank throughout the query builder and results list.

- 9. Close the query. Choose No when you are prompted to save the query.
- 10. Create a new Task work item. Choose the New Work Items link, and then choose Task.

This work item is created from the work item type that you changed and imported.

11. Notice, in the Status box, that the label for the renamed field, Rank, has not changed. This is because the field labels on the work item forms are scoped

to the parent team project and are independent of the server-wide field name just specified.

#### ⊠Note

For more information about how to change field labels on work item forms, see Change the Label for a Field.

12. Close the new Task and choose No when you are prompted to save the work item.

## **Change the report as value for a field**

The following command specifies the ability to report the type of the DateTime field AdventureWorks.CreatedOn to dimension. This field's data enters the warehouse and Analysis Services databases so that it can be used to filter reports.

witadmin changefield /collection:http://AdventureWorksServer:8080/t1

The following command specifies the ability to report the type of the Double field AdventureWorks.Field to measure. All measures are aggregated by sum.

#### 

witadmin reportfield /collection:http://AdventureWorksServer:8080/t1

#### **Enable synchronization of a custom person-name field**

The following command enables synchronization for the work item field AW.CustomerName defined for Collection1 on the AdventureWorksServer.

## Verify the data type of the field that you want to convert

• Verify the data type assigned to the field, such as

MyCompany.CustomerName, that you want to synchronize by entering the following command:

#### 

witadmin listfields /collection:http://AdventureWorksServer:808

## **Enable synchronization**

1. To enable synchronization for a person-named field, type the following command, substituting your data for the arguments that are shown here:

witadmin changefield /collection:http://AdventureWorksServer:80;

2. The following confirmation prompt appears:

# This will change properties for field {0} on the Team Foundation Server. Do you want to continue?

3. Type **0** to confirm that you want to change the field, or **1** to cancel the request.

If the change request succeeds, the following confirmation message appears:

#### The field was updated.

If the change request fails, an error message appears. The most common mistakes that can be made are trying to change a system reference field, or trying to change a field of a data type other than String. These operations are not supported.

## **Delete a field**

Before you delete a field, verify that the field is not in use. If the field is in use, you must first remove the field from the work item types that use it prior to

deleting it from the team project collection. The following command deletes the field AdventureWorks.Field from Collection1:

#### Copy Code

witadmin deletefield /collection:http://AdventureWorksServer:8080/t1

## Verify a field is not in use

1. Enter the following command, specifying the reference name for the work item field, such as MyCompany.MyProcess.MyField.

#### Сору

witadmin listfields /collection:http://AdventureWorksServer:808

In the information displayed for the field, verify that the value for **Use** is "**Not In Use**" as follows in this example.

#### Copy Code

Field: MyCompany.MyProcess.MyField Name: MyField Type: String Reportable As: dimension Use: Not In Use Indexed: False

2. If the **Use** field indicates that the field is in use, then you must delete it from each work item type for each team project that is listed. For example, the Microsoft.VSTS.TCM.SystemInfo field indicates that it is being used by the Bug and Code Defect work item types for four projects: Arroyo, Desert, Palm, and Springs.

#### Copy Code

Field: Microsoft.VSTS.TCM.SystemInfo Name: System Info Type: Html Reportable As: None Use: Arroyo (Bug), Desert (Bug), Palm (Bug), Springs (Bug, Code Indexed: False

Before you can delete this field, you must remove it from each of the work item types listed for each team project for which it is defined. To remove the field, you modify the definition for the work item type by deleting the FIELD and Control elements that contain the field reference name. See <u>Customize and Manage Work Item Types [witadmin]</u>, FIELD (Definition) Element, and Control XML Element Reference.

## Delete a field from a team project collection

• Enter the following command to delete the MyCompany.MyProcess.MyField field, and then choose Enter.

#### Copy Code

witadmin deletefield /collection:http://AdventureWorksServer:80;

Enter **y** at the confirmation prompt to complete this step.

## See Also

#### Concepts

Resolve Schema Conflicts That Are Occurring in the Data Warehouse Add and Modify Work Item Fields to Support Reporting Work Item Field Reference for Visual Studio ALM Reportable Fields Reference for Visual Studio ALM

#### **Other Resources**

witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Remove Work Items [witadmin]
See Also Send Feedback

You can permanently remove one or more work items from the Team Foundation database for a team project collection by using the **witadmin destroywi** command. Work items whose state is set to Closed remain in the database and can be reactivated. Permanently removed work items are removed from the database and cannot be restored nor reactivated.

Each work item represents an object that is stored in the Team Foundation database and that is assigned a unique identifier, which is referred to as a work item ID. Work item IDs are unique within a project collection.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

## Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace **%programfiles%** with **%programfiles(x86)%**.

Requirements

• You must be a member of the Team Foundation Administrators security group or the Project Administrators security group for the team project collection. See <u>Team Foundation Server Permissions</u> and <u>Team Foundation</u> <u>Server Default Groups</u>, <u>Permissions</u>, and <u>Roles</u>.

⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the shortcut menu for Command Prompt, and choose Run as Administrator. For more information, see the <u>Microsoft Web site</u>.

witadmin destroywi /collection:CollectionURL /id:id [/noprompt]

## **Parameters**

Parameter	Description
/ <b>collection</b> :CollectionURL	Specifies the URI of the team project collection. The the URI is the following: http://ServerName:Port/VirtualDirectoryName/Colle
	If no virtual directory is used, then the format for the following:
	http://ServerName:Port/CollectionName.
/ <b>id</b> :id	The ID of a work item to destroy. To specify multiple items, separate IDs using only commas, without white
/noprompt	Disables the prompt for confirmation.
/ <b>?</b> or <b>help</b>	Displays help about the command in the Command P window.

# **Examples**

## **Permanently Remove Work Items From the Database**

The following example deletes the work item 2003 from the database for Collection1 on the AdventureWorksServer server:

#### Copy Code

witadmin destroywi /collection:http://AdventureWorksServer:8080/tfs,

The following example deletes the work items with IDs, 12, 15, and 23 from the database for Collection1 on the AdventureWorksServer server:

### Copy Code

witadmin destroywi /collection:http://AdventureWorksServer:8080/tfs,

## See Also

#### **Other Resources**

Resolve, close, or reactivate a work item witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Rebuild the Client Cache [witadmin]
See Also Send Feedback

You can force a rebuild of the cache on each client computer the next time it connects to a team project collection by using the **witadmin rebuildcache** command.

To prevent workspace errors from occurring during version control or build operations in Team Foundation, the data cache on client computers must be updated after certain maintenance operations. After you move, restore, rename, or fail over a data-tier or application-tier server, you must refresh the cache for tracking work items and users must refresh the version control cache on client computers.

#### **▲Important**

To avoid server performance issues, you should not run this command during normal operating hours.

To run the **witadmin** command-line tool, open a Command Prompt window where either Visual Studio or Team Explorer is installed and enter:

#### Copy Code

cd %programfiles%\Microsoft Visual Studio 11.0\Common7\IDE

On a 64-bit edition of Windows, replace %programfiles% with %programfiles(x86)%.

Requirements

• To use the **witadmin rebuildcache** command, you must be a member of the Team Foundation Administrators security group or the Project Administrators security group for the project collection that you want to manage. <u>Team Foundation Server Permissions</u> and <u>Team Foundation Server Default Groups</u>, <u>Permissions</u>, and <u>Roles</u>.

#### ⊠Note

Even if you log on with administrative permissions, you must open an elevated Command Prompt window to perform this function on a server that is running Windows Server 2008. To open an elevated Command Prompt window, choose Start, open the shortcut menu for Command Prompt, and choose Run as Administrator. For more information, see the <u>Microsoft Web site</u>.

```
witadmin rebuildcache /collection:CollectionURL [/noprompt]
```

## **Parameters**

Parameter	Description
/ <b>collection</b> :CollectionURL	Specifies the URI of the team project collection. The the URI is the following: http://ServerName:Port/VirtualDirectoryName/Colle
	If a virtual directory is not used, then the format for tl the following:
	http://ServerName:Port/CollectionName.
/noprompt	Disables the prompt for confirmation.
/ <b>?</b> or <b>help</b>	Displays help about the command in the Command P window.

# **Remarks**

The **witadmin rebuildcache** command invalidates cached data on all clients for a specified team project collection. This causes the cache for each client to be refreshed the next time the client connects to the project collection.

# **Examples**

The following command invalidates the metadata cache for all clients that connect to DefaultCollection that is defined on the server that is named AdventureWorksServer. The client caches are updated the next time they connect to the project collection.

Copy Code

witadmin rebuildcache /collection:http://AdventureWorksServer:8080/1

## See Also

#### Tasks

Refresh Your Team Foundation Client

### Concepts

Choose the Team Foundation client to support your tasks

#### **Other Resources**

witAdmin: Customize and Manage Objects for Tracking Work Items

Send <u>feedback</u> on this topic to Microsoft.

Visual Basic C#
Visual C++
F#
HLSL
JScript
Add a Report Server to Your Deployment
Send Feedback

This topic has been merged into the tutorial <u>Add reports to a team project</u>.

•

Send <u>feedback</u> on this topic to Microsoft.