# PART I
# Getting Started

- **CHAPTER 1:** Introducing Team Foundation Server 2012
- **CHAPTER 2:** Planning a Deployment
- **CHAPTER 3:** Installation and Configuration
- **CHAPTER 4:** Connecting to Team Foundation Server

# 1

# Introducing Team Foundation Server 2012

**WHAT'S IN THIS CHAPTER?**

- Getting to know Team Foundation Server 2012
- Understanding what's new in Team Foundation Server 2012
- Acquiring Team Foundation Server 2012

This chapter introduces you to Microsoft Visual Studio Team Foundation Server 2012. Here you will learn what it is for, the key concepts needed when using it, and how to acquire it.

For those users already familiar with Team Foundation Server, the discussion in this chapter highlights areas that are new or have changed substantially. However, because understanding the legacy of a technology is always helpful, this chapter also includes some of the history of the Team Foundation Server product, which will help explain how it became what it is today.

This chapter also discusses the improved release model, including the ability to have Microsoft manage hosting, frequent upgrades, and backups by leveraging the Team Foundation Service. Later chapters will go into more depth with an examination of the architecture of the Team Foundation Server product.

## WHAT IS TEAM FOUNDATION SERVER?

Developing software is difficult—a fact repeatedly proven by how many projects have failed. An essential factor in the success of any software development team is how well the members of the team communicate with each other, as well as with the people who wanted the software developed in the first place.

Team Foundation Server provides the core collaboration functionality for your software development teams in a very tightly integrated product. The functionality provided by Team Foundation Server includes the following:

- Project management
- Work item tracking (WIT)
- Version control
- Test case management
- Build automation
- Reporting
- Lab and environment management
- Feedback management

Each of these topics is explored extensively in this book.

Team Foundation Server is a separate server product designed specifically for software engineering teams with developers, testers, architects, project managers, business analysts, and anyone else contributing to software development releases and projects. Logically, Team Foundation Server is made up of the following two tiers, which can be physically deployed across one or many machines:

1. Application Tier—The *application tier* primarily consists of a set of web services with which the client machines communicate by using a highly optimized, web service-based protocol. It also includes a rich web access site to interact with a server without having to install a client such as Visual Studio.
2. Data Tier—The *data tier* is made up of a SQL Server database containing the database logic of the Team Foundation Server application, along with the data for your Team Foundation Server instance. The data stored in the database is used by Team Foundation Server's reporting functionality. All the data stored in Team Foundation Server is stored in this SQL Server database, thus making it easy to back up.

Team Foundation Server was designed with extensibility in mind. It can integrate with a comprehensive .NET Application Programming Interface (API). It also has a set of events that allow it to integrate with outside tools as first-class citizens. The same .NET programming model and event system is used by Microsoft to construct Team Foundation Server, as well as the client integrations into Visual Studio.

Team Foundation Server has plenty of competitors, including other enterprise Application Lifecycle Management (ALM) systems and purpose-specific products (such as source control systems). The main benefit of having all the different systems available in one product is that Team Foundation Server fully integrates the different systems. This allows for true innovation in the development tools space, as you will notice with several of the new tools available in this latest release. Instead of having to worry about integrating the separate systems yourself, you can take advantage of the work that Microsoft has done for you.

**NOTE** *Jason Zander, Corporate Vice President for Visual Studio, makes this particular point well in a blog post originally about Team Foundation Server 2010. You can find the blog post at* `http://aka.ms/IntegratedALMSolution`.

When you compare enterprise ALM products currently on the market, you will discover that Team Foundation Server was designed to be easily customized and extended. Team Foundation Server ensures that developers using any development platform can participate and easily use Team Foundation Server, including Visual Studio, Eclipse-based development, Xcode, and many more.

## WHAT'S NEW IN TEAM FOUNDATION SERVER 2012

If you have used legacy versions of Team Foundation Server, you may be curious about what is new in the latest release. As this book demonstrates, it is a big release with considerable new functionality and improvements across the board. While many of these features are explained throughout this book, if you have used a previous version of Team Foundation Server, the features described in the following sections will be new to you. Some of the client-side topics are covered in more detail in the companion book to this volume, *Professional Application Lifecycle Management with Visual Studio 2012*.

### Version Control

One goal for this release was to dramatically improve the experience for developers and other team members who use the version control repository in Team Foundation Server. Many small features and improvements should be noticeable in reducing the friction of working with Team Foundation Server. For example, many of the algorithms used in the merge process have been improved so that you can trust the auto-merge process more, and fewer false positives show up as merge conflicts.

Another notable change for developers who have been using Visual Studio for a long time is the replacement for the Compare/Diff tool that has been around for a while. It now takes advantage of the new code editor in Visual Studio so that you can get IntelliSense, syntax coloring, etc., within the compare tool. There are also several different views you can use based on your preferences for reviewing the differences.

Included with all the small improvements in this release is a large improvement: local workspaces. Local workspaces removes the need to really understand workspaces before getting started and fundamentally changes how you work. Essentially, one of the frustrations of using Team Foundation Server in the past has been that files that have not been checked out have a read-only bit, which prevents someone from editing them without the explicit check-out. Local workspaces change this and "detect" changes automatically, thus removing the requirement for read-only bits. This allows you to use any editor, including Notepad, on any platform (Windows, Linux, Mac OS, etc.) to make your changes.

Some side benefits to the new local workspaces model include remarkable improvements in being able to work offline. Local workspaces and how this paradigm shift changes are both covered in detail in Chapters 5 and 6.

To date, the Team Foundation Server product team also introduced a fully supported tool for Git integration called `Git-tf.exe`. It allows a developer to create a local Git repository that is a clone of a central Team Foundation Server version control repository. This tool allows the team to work locally in that Git repository and then commit changes back up to the Team Foundation Server repository when they are ready. Because the command-line tool was implemented in Java, it can be used on several platforms. For example, an iOS developer can now use the native Git integration in the Xcode IDE on Mac OS and still be able to contribute to a Team Foundation Server repository as well, where other team members may be working within their organization.

## Code Reviews

Team Foundation Server 2012 also introduces a built-in set of experiences for requesting, responding, and managing code reviews. This set uses the powerful work item tracking experiences behind the scenes as well as some specialized user experiences to help you discuss changes. Opening a set of changes from a code review uses the new diff experience, as mentioned in the previous section.

You can also make comments at a code review, file, or specific set of lines of code-level, including the ability to have thread discussions. After you have finished completing a review, you can approve, send back with comments, or reject a code review.

## Managing Personal Work

New experiences are also available in Visual Studio 2012, when connected to Team Foundation Server 2012, for managing your personal context and work while developing. These new experiences start from the “My Work” pane in Team Explorer. Developers can indicate which task, bug, or work item they are working on and then check-in those changes together.

If developers are not finished with their work or are called to do something else, they can also suspend their work, which creates a shelveset. It exceeds the shelveset concept in previous versions of Team Foundation Server because it also stores the important context information about where the work was suspended. Developers can later “resume” their work and it will load the solution and projects, open windows, breakpoints, etc., back as they were when the work was suspended.

## Web Access

Team Web Access has been completely redesigned in Team Foundation Server 2012 to provide an even better experience for those without any of the traditional clients available. It is friendly to modern browsers, including mobile browsers, and works well with several form factors.

Some new features were implemented specifically in Team Web Access as well, including new administration portals and managing individual/team e-mail alerts. Team Foundation Server 2012 also introduces the concept of “teams” within a team project that can have team hubs dedicated to each of the teams. Teams are fully managed within Team Web Access.

## Project Management

In Service Pack 1 of Team Foundation Server 2010 and an associated feature pack in that release, the product team introduced support for bidirectional synchronization between project management work items in Team Foundation Server and tasks in Project Server. This synchronization is included in the Team Foundation Server 2012 release. You can learn more about Project Server integration in Chapter 15.

## Agile and Scrum Product Management

Additional new experiences added to Team Web Access are agile project management and product planning. The new Agile Planning tools are specifically designed for users practicing principles from Scrum, but can actually be beneficial for those using any process.

The primary tools introduced in this release are Product Backlog planning; Sprint/Iteration planning, including managing team capacity; live velocity and sprint burndown reports; and a new task board that can be useful for daily Scrums or status meetings. Each of these tools is team-oriented so that different teams can manage their own backlogs and progress.

One of the nice things introduced with these tools is the ability to track start and end dates on Iteration paths. These dates are then used around Team Foundation Server to provide a richer experience.

## Stakeholder Feedback

Teams and product owners can now request rich feedback from stakeholders, business users, and customers from working editions of their applications and software. Those stakeholders will receive an e-mail with a link to connect them automatically to Team Foundation Server and immediately start providing feedback using a new lightweight client appropriately named the Microsoft Feedback Client.

## Build

This release also includes several improvements to the built-in build system, including the ability to batch your gated check-ins. This can be useful if you receive numerous check-ins during the day that all need to be validated quickly. Additionally, the new Team Explorer includes some user experience improvements, including marking builds as personal or team favorites, and the ability to filter build definitions if your team project contains a lot of them.

Development teams that use third-party testing frameworks outside of `MSTest` will be happy to know that those teams can now include those tests in test runs during the automated build process in Team Foundation Server 2012.

# ACQUISITION OPTIONS

Microsoft also greatly improved how you may acquire Team Foundation Server. Several options are available to you, as discussed in the following sections.

Licensing can be somewhat confusing, but Team Foundation Server licensing follows the licensing pattern of other Microsoft server products. There is a server license. Additionally, with some notable exceptions, each user that connects to the server should have a Client Access License (CAL) for Team Foundation Server.

> **NOTE** *For more information about those potential exceptions, or questions about what you will need to purchase, you can seek help from a Microsoft Partner with the ALM Competency, your local Microsoft Developer Tools Specialist, or reference the End-User License Agreement (EULA). A licensing white paper dedicated to Visual Studio, MSDN, and Team Foundation Server is also available here:* `http://aka.ms/VisualStudioLicensing`.

## Team Foundation Service

By far, the easiest way to get started with adopting Team Foundation Server is through a new hosted option available directly from Microsoft called the Team Foundation Service. It shares a majority of the same code base as the same Team Foundation Server product used on-premises but modified to be hosted from Windows Azure for multiple tenants. It is available at `http://tfs.visualstudio.com`.

The best part of using the Team Foundation Service is that your team need not worry about backups, high availability, upgrades, or other potential time-consuming administration and maintenance tasks. Another nice thing is that Team Foundation Service customers, before on-premises customers, will receive frequent updates that even include new features.

> **NOTE** *Brian Harry announced that the internal product teams improved their engineering process so well over the past two to three years that they are able to quickly provide more frequent updates. Starting with Team Foundation Server 2012, the product team is planning to provide updates that include the typical performance and bug fixes but also brand new features. The frequency is still to be determined, but the first on-premises update will be available before the end of calendar year 2012.*
>
> *Team Foundation Service customers will see updates made more frequently than the on-premises edition. Brian mentioned that his teams are able to deploy hotfixes daily, but plans to provide full-featured updates every three weeks, which lines up with the internal sprint schedule. You can learn more about this topic from Brian Harry's blog post at* `http://aka.ms/TFSReleaseCadence`.
>
> *One thing to take away from this discussion is to make sure that your team always uses the latest update of Team Foundation Server if you choose to install it on-premises.*

Teams using the Team Foundation Service are able to leverage an elastic set of standard build servers. This elastic build service provides standard build machines available and clean for each of your builds. Teams can even integrate their elastic builds with their Windows Azure accounts to provide continuous deployment to instances of their applications or sites hosted in Windows Azure. Teams can also take advantage of on-premises build servers connected to a Team Foundation Service.

Microsoft has released the Team Foundation Service for general availability in late 2012. They have announced that the full feature set will be provided to teams of up to five at no cost. Additionally, MSDN subscribers will be able to leverage Team Foundation Service as an additional benefit to their MSDN subscription. If you or your team do not fall into either of those two categories, you will be able to leverage the Team Foundation Service without any costs until more pricing details are announced. At that point, if you decide that you no longer want to continue to use the Team Foundation Service, you are free to download your source code and close your account.

To date, the Team Foundation Service does not have full parity with the on-premises product. For example, the lab management, reporting, and process template customization capabilities are features not currently available. As the Team Foundation Service evolves over time, there will be greater, if not full, parity with the on-premises edition.

In the meantime, for teams that would like the full set of features but still have someone else manage their Team Foundation Server instance, options are available through several third-party hosting companies.

## Express

Starting with Team Foundation Server 2012, small software engineering teams can leverage an Express version of Team Foundation Server 2012 that is available and free for up to five developers. Team Foundation Server Express is available at `http://aka.ms/TFS2012Express`. The Express edition includes, but is not limited to, the following core developer features:

- Version Control
- Work Item Tracking
- Build Automation

This is a perfect start for small teams that want an on-premises Team Foundation Server instance without any additional costs. If your team grows beyond five, you can always buy CALs for users six and beyond. The Express instance can even be upgraded at any time to take advantage of the full set of features without losing any data.

## Trial

One of the easiest ways to acquire Team Foundation Server is on a 90-day trial basis. You can download the full version of Team Foundation Server and try out all of the features without having to purchase a full copy. The DVD ISO image for the trial edition is available at `http://aka.ms/TFS2012Downloads`.

If you install the trial edition of Team Foundation Server, you can easily apply a product key to activate the trial edition. You could even move the team project collection from the trial server to a different server instance once your team has decided to fully adopt Team Foundation Server.

Alternatively, if you need a 30-day extension, you can perform one extension using the Team Foundation Server Administration Console once the time gets near the end of the trial period. You can find out more information about extending the trial by visiting `http://aka.ms/ExtendTFSTrial`.

If you would rather have a virtual machine that is ready to use (including all of the software necessary to demo and evaluate Visual Studio 2012 and Team Foundation Server 2012), you can download the all-up Visual Studio 2012 virtual machine image. The virtual machine has a time limit that starts from the day that you first start the machine. You can always download a fresh copy of the machine to begin your demo experience over.

> **NOTE** *You can find the latest version of the virtual machine available at any time at* `http://aka.ms/VS2012VirtualMachine`.

## Volume Licensing

Microsoft has plenty of options for volume licensing, including Enterprise, Select, Open Value, and Open License Agreements, that will help your company significantly reduce the overall cost of acquiring an on-premises edition of Team Foundation Server. Different options are available based on your company size and engineering team size. This option is by far the most popular choice for companies looking to acquire Team Foundation Server, MSDN subscriptions, and Visual Studio licenses.

If your company acquired an earlier version of Team Foundation Server through a volume licensing program, and also purchased Software Assurance (SA), you may be entitled to a license of Team Foundation Server 2012 without additional cost, if the SA was still active on the date that Team Foundation Server 2012 was released.

> **NOTE** *For more information about volume licensing, discuss your options with your Microsoft internal volume licensing administrator, your local Microsoft Developer Tools Specialist, or a Microsoft Partner with ALM Competency. You can find out more information from the Visual Studio Licensing white paper available at* `http://aka.ms/VisualStudioLicensing`.

## MSDN Subscriptions

Beginning with the Visual Studio 2010 release, which now continues for the Visual Studio 2012 release, a full production-use license of Team Foundation Server 2012 is included with each license of Visual Studio that includes an MSDN subscription. Those MSDN subscribers also receive a Team Foundation Server 2012 CAL available for production use.

This now enables developers, testers, architects, and others with an active MSDN subscription to take advantage of Team Foundation Server without additional licensing costs.

**NOTE** *For more information about MSDN subscriptions and for links to download Team Foundation Server 2012, visit the MSDN Subscriber Downloads website at* `http://msdn.microsoft.com/subscriptions`.

## Microsoft Partner Network

Companies that are members of the Microsoft Partner Network and have achieved certain competencies can be entitled to development and test-use licenses of several of the products included with an MSDN subscription, including Team Foundation Server 2012.

**NOTE** *For more information about the requirements and benefits available for Microsoft Partners, you can visit* `http://partner.microsoft.com`.

## Retail

If you are not able to use any of the other acquisition methods, you can always acquire Team Foundation Server 2012 through retail channels, including the online Microsoft Store. You can purchase the product directly from Microsoft online at `http://aka.ms/TFS2012Retail`. It is also available from many other popular retail sites.

One of the nice benefits of purchasing a server license using the retail channel is that you also receive a CAL exclusion for up to five named users. This benefit is available only from licenses purchased through the retail channel, and it is not included with other acquisition avenues discussed in this chapter.

# SUMMARY

As you learned in this chapter, Team Foundation Server is a product with lots of features and functionality. This chapter introduced the types of features available, including those new to the latest release. Additionally, you learned about the different acquisition methods for getting the software for Team Foundation Server.

The next few chapters will familiarize you with planning a Team Foundation Server deployment, installing a brand-new server, and then the different methods available for connecting to your new server. Chapter 2 begins that discussion with an examination of deploying Team Foundation Server.