This book starts with giving the reader a walkthrough of TFS and the reporting architecture, guiding you to master the comprehensive reports that are part of Team Foundation Server.

You will explore the Team Project Scenario with work item queries, flat queries, direct link queries, and tree queries. Moving ahead, you will learn about the Excel reporting generation process and how to create ad hoc queries and reports in Microsoft Excel to help manage a team project. Microsoft Excel allows users to connect to the TFS Data and display results as a PivotTable or PivotChart report.

Then the book describes each of the standard report dashboards such as SharePoint Dashboard, Project Dashboard, Progress Dashboard, Quality Dashboard, Test Dashboard, Bugs Dashboard, and Build Dashboard. Finally, you will learn how to use the powerful tools available in SQL Server Reporting Services to create and customize robust reports and also learn how to design and customize a dashboard of reports relevant to you.

Who this book is written for

This book is intended for developers, testers, architects, and project managers who want to explore and make use of the reporting facilities of Team Foundation Server 2013. Although no previous experience of reporting is required, a basic understanding of the Team Foundation Components and project templates would be a plus.

What you will learn from this book

- Get to know about TFS 2013, Reporting Architecture, and Application Lifecycle Management (ALM)
- Generate work item reports
- Explore Excel standard reports (bugs, tracking, and test)
- Create reports based on SQL Server data using Report Designer
- Manage SSRS reports in Report Manager
- Display visual reports as charts and graphs in Team Web Access

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Team Foundation Server Primer'
- A synopsis of the book’s content
- More information on Reporting in TFS

About the Author

Dipti Chhatrapati is a Microsoft SharePoint specialist and is currently working as a senior consultant at Capgemini in Mumbai, India. She has more than 7 years of experience in application development, design, maintenance, and administration and is continuously working to improve her skills and keeping them up to date. As a developer and project leader, she has been an extensive user of Microsoft Visual Studio and Team Foundation Server throughout her career. Dipti blogs at www.sharepointrun.com and can be reached on Twitter at @dips84 and via e-mail at diptichhatrapati@gmail.com. Dipti is very passionate about music, sports, and arts, and loves to spend her free time learning new things, socializing with friends and family, cooking, exercising, and singing.
Reporting in TFS

This book is your guide to mastering the comprehensive reports that are part of Team Foundation Server and Microsoft's main Application Life Cycle Management (ALM) suite. TFS Reporting provides the ability to track different metrics of development projects throughout the Application Life Cycle; these include team progress, bugs, the current backlog, test progress, and so on, using well-integrated and familiar tools such as Excel and SQL Server Reporting Services. Team Web Access reports are also explained in this book, providing a quick progress status view via different charts, work item queries, as well as inbuilt reports. The book starts by giving you a walkthrough of TFS and the reporting architecture. Then it jumps into the basics of Excel reporting, describing each of the standard reports available. Finally, you will learn how to use the powerful tools available in SQL Server Reporting Services to create and customize robust reports and how to design and customize a dashboard featuring reports relevant to you.

What This Book Covers

Chapter 1, Team Foundation Server Primer, teaches the basics of the TFS architecture and reporting, and the importance of Application Life Cycle Management.

Chapter 2, Work Item Querying, explains the team project scenario in TFS walks through the types of work item query that results the required work items to know the status of work progress.

Chapter 3, Excel Reporting, explains the basics of integrating SharePoint 2013 with TFS 2013. This chapter also explores Excel report types and how to customize them as required need on different types of dashboards.

Chapter 4, SQL Server Reporting, walks through creating basic SQL reports using Report Builder and Report Designer.

Chapter 5, Team Web Access Reporting, explains Team Web Access levels and their various features (as required) and walks through a very light feature called work item charting. Also, it describes inbuilt standard reports.
**Team Foundation Server Primer**

**Team Foundation Server** (TFS) is Microsoft's main tool to maintain development projects within the project team. It provides the collaborating platform for a software product. TFS also helps with maintaining source code with check-in/checkout and code review functionalities, building and testing the application, planning and tracking the work of team members, and so on. TFS is available as part of Microsoft's primary development suite, **Visual studio**. It is available on various Visual Studio editions such as Professional, Premium, Ultimate, and Test Professional. This chapter promotes a basic understanding of TFS, **Application Lifecycle Management** (ALM), and the TFS 2013 architecture.

**Chapter objectives**

This chapter covers the following topics:

- The basics of TFS
- ALM and TFS 2013
- The TFS 2013 architecture
- TFS 2013 and reporting
TFS basics

TFS is a set of tools and processes that are used to plan, develop, build, test, deploy, and maintain a software product. TFS forms the basis for Microsoft's Application Lifecycle Management (ALM) initiative. It also has a cloud version in Azure named Visual Studio Online (VSO), which is known as TFS Online and has slightly different capabilities. VSO is a hub for project data on cloud, which can be up-and-running in a few minutes without installing and configuring a single server. We can connect to the project on cloud using the Visual Studio Development suite.

TFS ensures that the various software development methodologies and activities of an application include the following:

- Connecting the team
- Version Control
- Planning Agile Processes
- Maintaining test cases
- Reporting the status of the project

Connecting the team

The main purpose of TFS is to empower a team with easy-to-use methods and tools to develop software solutions. It enhances communication between team members, including developers, testers, the project manager, scrum master, and product owners, but also eases communication for external stakeholders by reporting all kinds of aspects within the development process such as bug reports, the results of testing, maintenance reports, work item reports, and so on. Everything in TFS is done within the context of a team project. The team project is used to maintain the project's development work where members are added and given access through which reports are generated and displayed. The following diagram indicates the connection between the project team and the TFS Server.
Version control

TFS provides version control to enable a team to manage and track the source files and documents in a project. It lets you to add not only code files but also documents, to maintain the project's documentation.

TFS version control has the following most useful activities:

- Check in/check out
- Managing versions
- Maintaining control of team projects
- Branching
- Shelving
- Defect tracking

Microsoft recommends using version control as early as possible in the project development life cycle. It helps to improve productivity and get rid of difficult problems.

For more information on Version control, have a look at the following links:


Planning the Agile process

TFS supports agile methodologies by providing multiple process templates for agile projects and scrums. Third-party process templates can also be used to create your own template. Such templates help to plan, manage, and track a project team's work activities. The following are the main features provided by TFS for agile project development:

- Managing product backlogs
- Creating work items
- Breaking work items into tasks
- Assigning tasks to team members
- Tracking processes via a task board
In short, TFS provides real-time visibility with customizable boards while continuously monitoring the project status with flexible reporting. Agile tools are also available on Visual Studio Online.

For more information on agile process planning, have a look at the following link:


### Processing build

TFS allows continuous or periodic generation of project builds. A TFS build consists of the following activities:

- Synchronizing sources
- Compiling applications
- Running unit tests
- Performing code analysis
- Releasing builds on a file server
- Publishing build reports
- Scheduling a build
- Code check-in after build

In reality, TFS build server provides a way to define a build, which is then compiled and tested. Build systems help catch bugs and other quality issues during the development period and keep the code validated so that the application performs better.

For more information on the build process, have a look at the following links:

Maintaining test cases
The TFS family has a tool named Microsoft Test Management that is used to store test plans, create test cases, and run these test cases on a regular basis to increase the quality of the software product. It also helps find bugs encountered in an application. Generally, Microsoft Test Manager defines the test plan and manages the test cases for manual or automated tests. As we saw previously, test cases can be run during a build process; test cases are actually stored in a TFS server and closely integrated with every build process throughout the software product life cycle.

For more information on test case management, have a look at the following links:


Reporting status
Reporting is a basic segment in TFS that uses a separate data warehouse to generate the report. Tracking a team's work via reporting using work items, queries, builds, source code, and test results, is mainly used to represent graphical charts in a TFS report. Based on this report, we can make better decisions in order to drive the project seamlessly. There are a number of templates available to create reports. TFS can create custom reports as well. Every report depicts the status of the project, quality of the software, or progress of the project, which in turn results in better management of the software product throughout the team. We will discuss TFS reporting in more detail in later chapters.

Ultimately, TFS is a focal point for the Application Life Cycle Management (ALM). It empowers the development team, project managers, and customers to effectively engage in the development process via a single solution.
ALM and TFS 2013

Application Life cycle Management (ALM) manages a project's life cycle in order to improve efficiency and reduce risk. It's basically a strategy that describes how to develop the code and work together. Application Lifecycle Management (ALM) is a set of proven practices that are used to manage the development process of the project that can be achieved via the Visual Studio development suite. These are the specific tools that are used to understand the customer’s needs, and hence the team can effectively design, implement, and deploy the code.

Wikipedia Definition: Application lifecycle management (ALM) is the product lifecycle management (governance, development, and maintenance) of application software. It encompasses requirements management, software architecture, computer programming, software testing, software maintenance, change management, project management, and release management.

ALM practices

ALM's core practices are as follow:

- Building the TFS Environment setup
- Creating a team project in TFS
- Adding team member accounts in a team project
- Sharing the code using version control in TFS
- Planning project work items
- Tracking the team's work
- Generating reports
- Setting up a build server
- Defining build processes
- Testing an application

The latest features of TFS 2013 and VSO empower team members to develop and manage the project that result in the best experience across the platforms. Moreover, additional ALM features help team members become more productive with improved support of agile software development practices. Most businesses require flawless synchronization of data across different ALM tools/solutions without losing focus on the value of a corporate ALM strategy.
Visual Studio Online (VSO) is also an ALM tool hosted in the cloud that has all the features of an on-premise TFS, minus the administration headache associated with an on-premise version.

Visual Studio Online (VSO) is a set of essential components on a cloud-based service to write better code. It offers code browsing, code editing, repository places to check-in /check out your code, build services, test case management, and application insights.

Using Visual Studio Online (VSO), you can create team projects, choose a process template that suits your project, and perform all the workflows associated with Sprint Planning and Project Management using the Agile methodology. The core capability of TFS 2013 is ALM, which has been evolving through various practices, tools, and processes, as depicted in the following:

TFS 2013 is the first requirement in setting up the environment in order to follow Application Lifecycle Management (ALM) practices and tools. It's designed to integrate all artifacts across the software development life cycle. It simply includes managing the source control and tracking work items that can be everything from requirements to bugs. All these things can be wrapped using process templates such as Scrum, CMMI, or custom ones, as required.
ALM with business trends

These days, business requirements have been changing to implement the business application from a single server to a hybrid server or completely on-cloud across various devices. The following screenshot represents some of the latest business trends that can be fulfilled via the Application Lifecycle Management (ALM) process:

By considering the latest business trends, ALM focuses more on building the application, measuring the application, and learning the application; this is called as the BUILD-MEASURE-LEARN cycle.

As we develop software, we get continuous feedback that goes to the project team and supplies insight into improving the project. Along with this, ALM can be broken down into four major areas: plan, develop, release, and operate, that continuously adds value to the solution, as shown in the following screenshot:
The current business scenario indicates that delivering high-quality, modern applications requires modern application lifecycle management (ALM) tools and processes. ALM offers processes and tools to enable software development teams to be more productive and collaborate more effectively with all project team members.

For more information on Application Lifecycle Management (ALM), have a look at the following links:

The TFS 2013 architecture

TFS 2013 is built on multilevel scalable architecture. To set up the TFS environment for the project team, it’s very important to figure out the architecture of TFS in order to understand the business need deployment strategy.

TFS can be deployed in three ways:

• On-premises with a single server or multiple servers across one domain or workgroup or domains
• On-cloud, where all deployments are server-hosted by Microsoft
• Hybrid Deployment, with TFS features as well as Visual Studio Online (VSO)

Based on business requirements, TFS architecture topology can be decided by considering the following:

• The Application, Data, and Client tiers
• The location of servers
• TFS Build computers and their locations
• Their necessity for a TFS server proxy
• Ensuring client access for the service on port 443
• Web services, databases, and object models
• Default TFS ports and protocols
• Permissions on TFS components

TFS on-premises

With TFS on-premises, we can extend the features and functionalities of the Team Foundation by writing the application either on the client side or on the application server. The TFS architecture is mainly divided into 4 parts:

• Client side: By using a client object model, we can extend TFS capabilities that relate to version control, tracking work items, and building on the client side.
• Application tier: By using a server object model, we can extend TFS functionalities on an application tier that relate to integrating other tools and data into TFS.

• Data tier: This includes data, stored procedures, and other associated logic. The data tier consists of the following databases within a SQL server:
  - Configuration database
  - Application warehouse
  - Analysis service database
  - Team project collection database

• Build machine: We can customize the build process using the build process object model on a build machine that relates to creating build processes and activities, as shown in the following TFS Architecture diagram:
For instance, TFS server hosts multiple team projects and each project will have a repository that will have four branches such as Development, **System Integration Testing (SIT)**, **User Acceptance Test (UAT)**, and Production, as shown in the following team project hierarchy:

![Team Project Hierarchy Diagram]

The code in the repositories resides in the data tier, which is accessible using a client application stored on the client side through the application tier. For each branch, you can configure one or more multiple build processes. These builds requires a build server. We can have one build server for all the builds or a separate build server for each repository; alternatively, it can be share a build server between two repositories.
Visual Studio Online

Microsoft also came up with the choice of using Visual Studio Online (VSO), which facilitates all server-side deployments. For example, all source code, work items, build configuration, and team features are hosted on-cloud by Microsoft. This feature immensely simplifies deployment, as we only need to bother with client-side components and Internet access.

In order to access cloud-based TFS, we require a Microsoft account to connect to the service on-cloud using a web browser. Using Visual Studio Online (VSO), we can perform activities such as creating team projects, adding members to the team, and working as you wish, as you do for locally deployed TFS; this is like getting rid of the administering servers. The following screenshot shows the flow of Visual Studio Online (VSO) and Team Project:

In Visual Studio Online (VSO) deployment, the application tier, data tier, and build server are hosted on the cloud via the Microsoft Cloud platform and SQL server Azure technology.
Hybrid deployment

However, there are still arguments for having TFS on-premise and Visual Studio Online (VSO). The following points are the main differences between TFS on-premises and TFS on-cloud:

<table>
<thead>
<tr>
<th>Feature</th>
<th>TFS on-premise</th>
<th>TFS on-cloud (VSO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customized work items and process templates</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>SharePoint and Project server Integration</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Data warehousing and Reporting service</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>CodeLense Support</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Active Directory Support</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Data availability within the network</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Application Insights</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Cloud Load Testing</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Always on the latest version</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Simple installation</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Virtual Team Rooms</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Supports Microsoft IDs</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>

Team Foundation Server has customizable work items and process templates, enterprise-level reporting, and incredible scalability. Visual Studio Online (VSO) has elastic load testing, Application Insights, and deep integration with Azure. It looks as if enterprises have to choose one or the other, but it's not always true. We can have the advantage of both to add more power to Application Life Cycle Management.

There are various integration tools available for TFS and Visual Studio Online (VSO). All are open source and you can choose one as per your requirements; they are as follows:

- CodePlex Integration Tools
- Team Foundation Server Integration Tools
- ALM Rangers
For more information on the TFS 2013 architecture, have a look at the following links:

- [http://visualstudiogallery.msdn.microsoft.com/eb77e739-c98c-4e36-9ead-fa115b27fefe](http://visualstudiogallery.msdn.microsoft.com/eb77e739-c98c-4e36-9ead-fa115b27fefe)

**TFS and reporting**

TFS 2013 has a reporting service that you can use to design reports; you can display them on the dashboard in order to show the team's progress in the project's development life cycle. As this book's main focus is on TFS reporting, it's important to understand how data flows in TFS for the reporting service and its architecture.

**The reporting architecture**

The reporting service is integrated with a SQL server as it has a separate data warehouse to generate reports. The main component of the TFS reporting architecture are:

- **Operational Stores**: Every activity in TFS gets stored in relational databases in the SQL server, these are known as operational stores. These mainly contain the TFS configuration and team project collection information.

- **Warehouse adapters**: These are manageable assemblies that fetch data from operational stores and transform them into a standardized format that is compatible with warehouse databases and writes them to the warehouse relational databases.

- **Relational Databases**: These have a schema that specifies the fields such as dimensions, measures, and reporting data details. The data warehouse is organized in a star schema, consisting of fact tables and dimension tables. For example, the work item fact table has one row for every work item stored in the work item operational store. Fact tables are a good source of information for the reports and show the latest activity. A dimension table stores the set of values that exists for given dimensions. Measures are the values taken from operational data.
• **Analysis service cube:** To report about day-to-day work items or test results, the warehouse needs to retain the state of every item for each day; this allows the data cube to aggregate the measures by day. The data cube aggregates the data from the relational databases. Whenever the data cube is processed, the data from the relational databases is pulled into the cube to get aggregated and stored. The cube provides a central place to obtain data for reports without having to know the schema for each operational store and access each store separately.

• **Reporting with Report Designer:** This is a tool in Visual Studio that is used to design the reports. Report Designer provides tabbed windows for data, layout and preview, query builder, and an expression editor to create a report.

• **Excel Reports:** TFS is integrated with Excel to manage projects and its reports that use Microsoft Excel. Microsoft Excel provides pivot tables and charts to view and analyze multidimensional data. You can bind these pivot tables directly to the Team Foundation cube, so you can interact with the data in the cube.

• **Security:** TFS administrators have access to the data in the data warehouse by granting or revoking permissions on the user's account. By default, write access to the warehouse is restricted to a service account under which the warehouse service runs. A user who has permission to view the data in the warehouse has full access to all of the data for all team projects in all team project collections.

The following diagram shows the reporting architecture for TFS 2013:
For more information on TFS and Reporting, have a look at the following links:


Types of reports
TFS 2013 offers the following types of report, which we will look at in depth in later chapters:

- **Work item queries**: These are used to review or update work items.
- **Work item reporting**: Custom work item reports are generated using Microsoft Excel, the project server, and SSRS reporting tools.
- **Excel reports**: There are a number of standard Excel reports available via the selected process template while creating a team project; these reports can be accessed via the SharePoint portal or Team Explorer's **Document** tab.
- **SSRS reports**: There are a number of default reports available via the selected process template with installed SSRS; these can be accessed via Web Access, SharePoint or Team Explorer.
- **Team Web Access reports**: TFS 2013 has excellent chart and standard reports that show a visual representation of work items. It can be accessed via Team Web Access.
- **Light weight reports**: TFS 2013 introduced a new feature called **light weight reporting** that provides the ability to create real-time reports based on query results and does not rely on the warehouse or cube. It offers real-time Burndown charts, velocity, and CFD diagrams directly within Team Web access.
• **SQL queries**: Major reports in TFS display information from the analysis service OLAP cube. However, running T-SQL queries directly against relational databases provides the facility to create more granular-level custom reports.

• **TFS API**: This provides binaries to create a custom report using only a few lines of C# code.

• **REST API**: Currently, this API is only available in Visual Studio Online (VSO) and is an implementation of OData Protocol. The REST API makes it dramatically easier to integrate Visual Studio Online (VSO) with other web-based tools and to access VS Online from any mobile device, irrespective of whether they are third-party commercial tools or custom ones you are building.

**Summary**

In this chapter, we reviewed the basics of TFS Architecture and Reporting. We also reviewed the importance of Application Life Cycle Management. In the next chapter, we will emphasize on creating work item reports from queries.
Where to buy this book

You can buy Reporting in TFS from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.