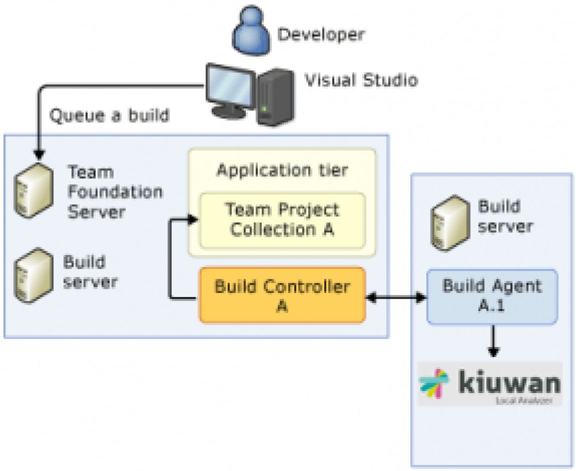
Continuous Inspection with Team Foundation Server

This guide will show you how to integrate Kiuwan inspections in your software development cycle using Team Foundation Server.

Contents:

Requirements

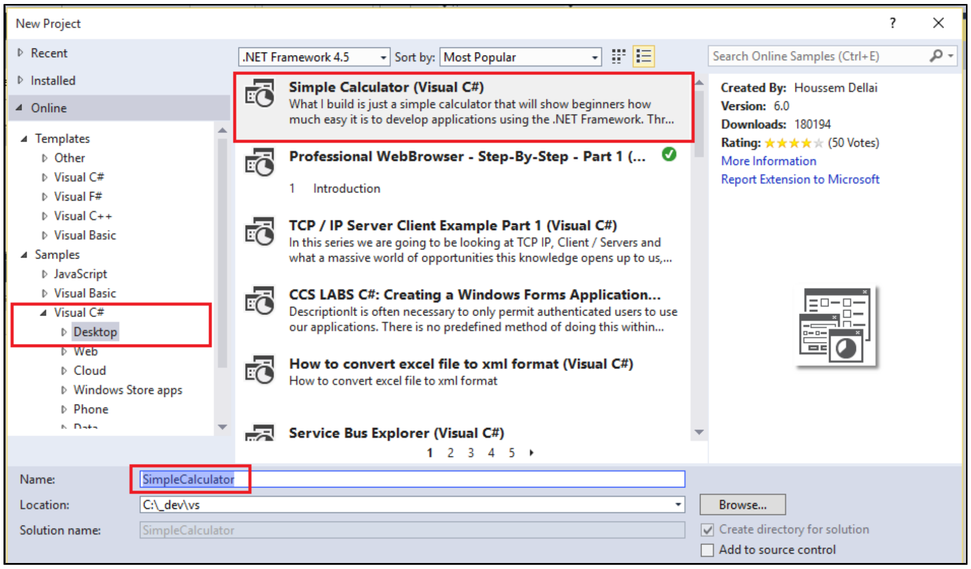
Requirement	Description	Image
Visual Studio 2013 and Team Foundation Server 2013	Both installed "on-premises".	
Kiuwan Local Analyzer	Installed on every machine where a TFS Build Agent is installed. Log into your Kiuwan Account via the KLA to ensure a working connection. Your credentials will be saved in a cipher form for subsequent analysis when they are run from the command line interface as well. Assuming kiuwan has been installed in c:\, we need to create a small script that will be invoked from the TFS build agent.	 Developer Visual Studio Queue a build Team Foundation Server Application tier Team Project Collection A Build server Build Controller A Build server Build Agent A.1 kiuwan Local Analyzer

```
@echo off
:: tfs2kiuwan.cmd
:: script to launch kiuwan analysis from team foundation server build.
:: -------------------------------------------------------------------
setlocal

set KIUWAN_HOME=C:\KiuwanLocalAnalyzer
set KIUWAN=%KIUWAN_HOME%\bin\agent.cmd

%KIUWAN% -c -n "%TF_BUILD_BUILDDEFINITIONNAME%" -l "%TF_BUILD_BUILDNUMBER%" -s "%TF_BUILD_SOURCESDIRECTORY%"

endlocal
```

Sample Application	Go to Visual Studio > Help > Samples to download the sample application for this example.	

New Project
Recent
Installed
Online
Templates
Other
Visual C#
Visual F#
Visual C++
Visual Basic
Samples
JavaScript
Visual Basic
Visual C#
Desktop
Web
Cloud
Windows Store apps
Phone
.NET Framework 4.5
Sort by: Most Popular
Search Online Samples (Ctrl+E)
Simple Calculator (Visual C#)
What I build is just a simple calculator that will show beginners how much easy it is to develop applications using the .NET Framework. Thr...
Professional WebBrowser - Step-By-Step - Part 1 (...
1 Introduction
TCP / IP Server Client Example Part 1 (Visual C#)
In this series we are going to be looking at TCP IP, Client / Servers and what a massive world of opportunities this knowledge opens up to us,...
CCS LABS C#: Creating a Windows Forms Application...
DescriptionIt is often necessary to only permit authenticated users to use our applications. There is no predefined method of doing this within...
How to convert excel file to xml format (Visual C#)
How to convert excel file to xml format
Service Bus Explorer (Visual C#)
1 2 3 4 5
Created By: Houssem Dellai
Version: 6.0
Downloads: 180194
Rating: ★★★★★ (50 Votes)
More Information
Report Extension to Microsoft
Browse...
☑ Create directory for solution
□ Add to source control

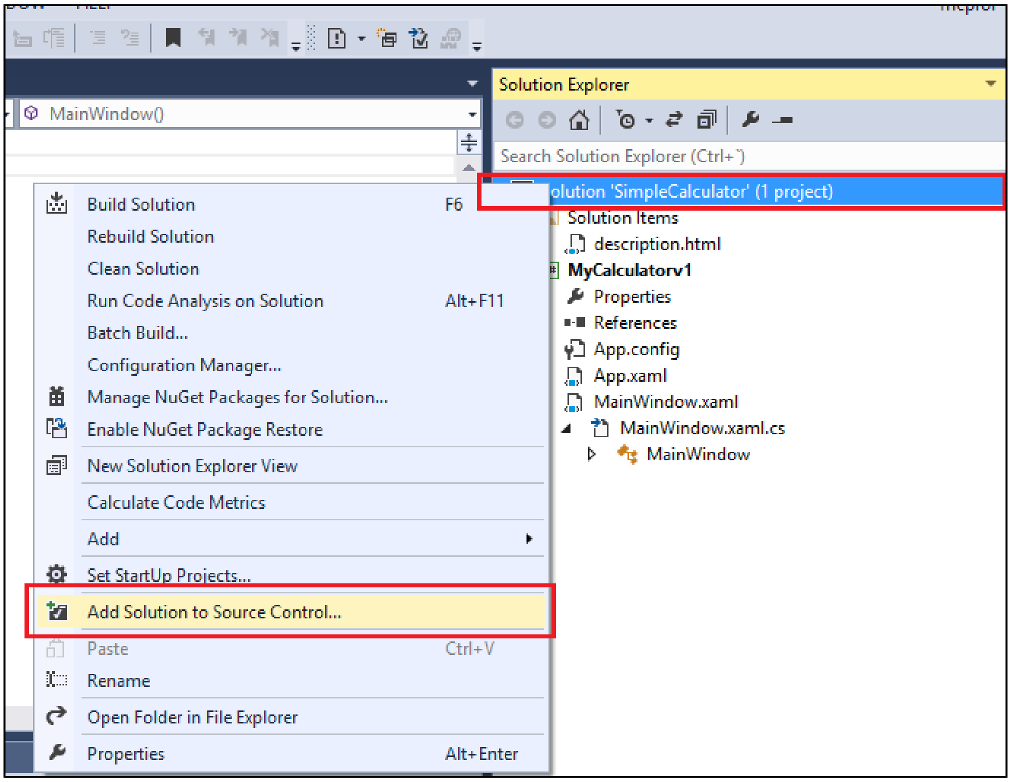
Name: SimpleCalculator
Location: C:_dev\vs
Solution name: SimpleCalculator

1. Download sample application

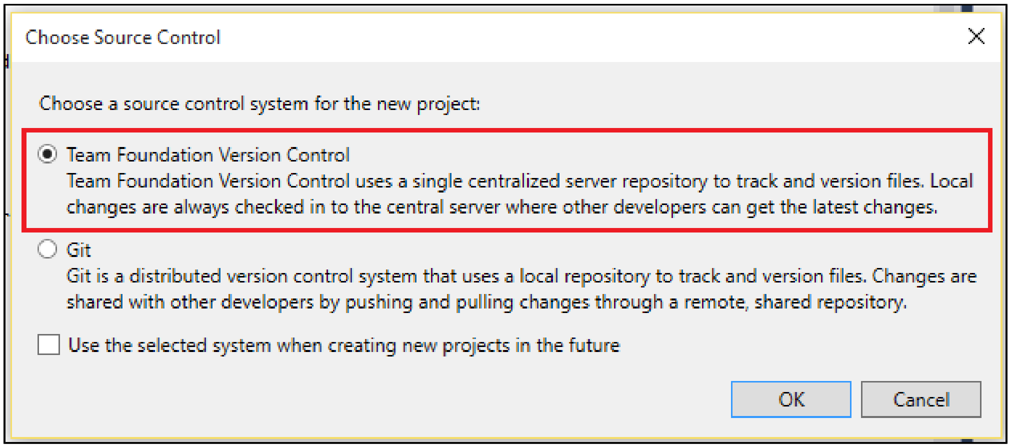
Download the sample application as described in the requirements.

2. Add the application to Source Control

In **Solution Explorer**, right-click on the **SimpleCalculator** solution. Then go to **Add Solution to Source Control**.



A new dialog appears: **Choose Source Control** backend. Select **Team Foundation Version Control**.



Choose Source Control

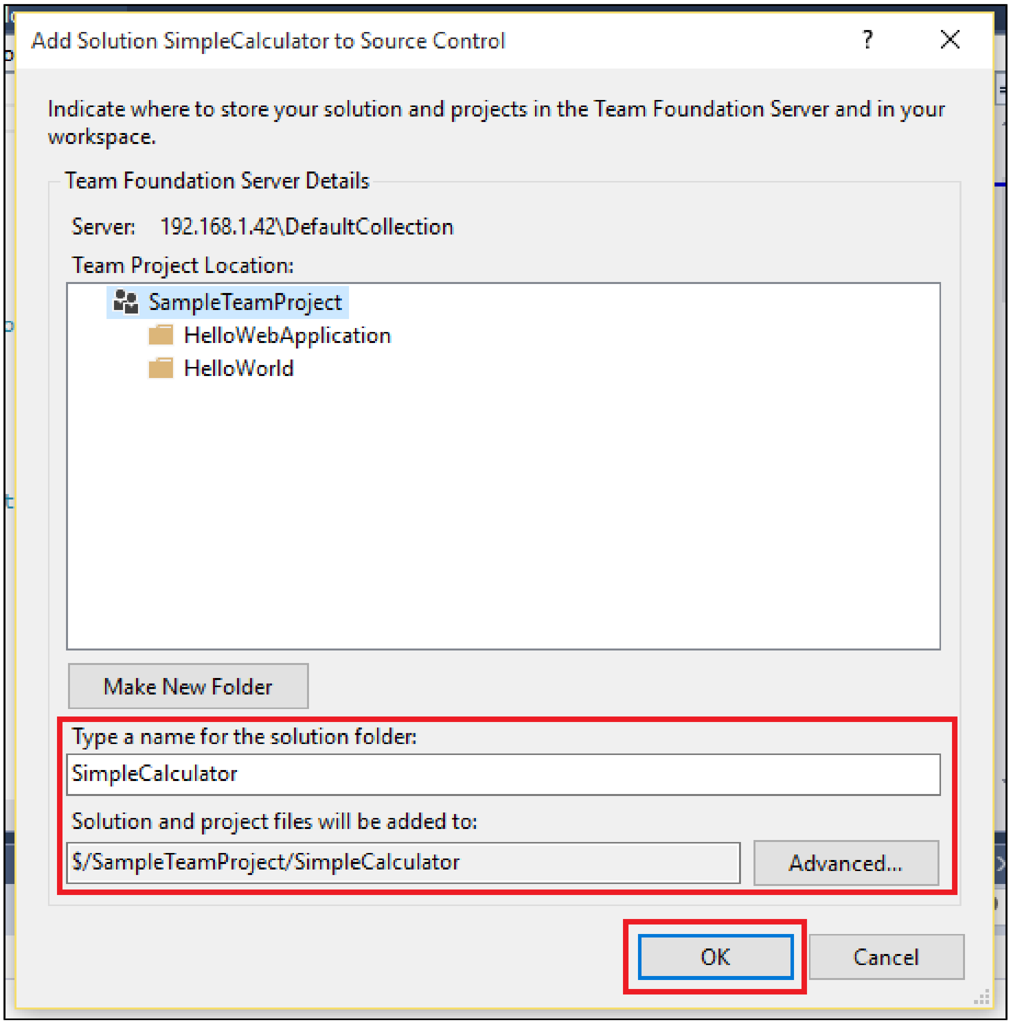
Choose a source control system for the new project:

● Team Foundation Version Control
Team Foundation Version Control uses a single centralized server repository to track and version files. Local changes are always checked in to the central server where other developers can get the latest changes.

○ Git
Git is a distributed version control system that uses a local repository to track and version files. Changes are shared with other developers by pushing and pulling changes through a remote, shared repository.

□ Use the selected system when creating new projects in the future

Finally, select a **TeamProject** where to place this project and create the solution folder.



Add Solution SimpleCalculator to Source Control

Indicate where to store your solution and projects in the Team Foundation Server and in your workspace.

Team Foundation Server Details

Server: 192.168.1.42\DefaultCollection

Team Project Location:

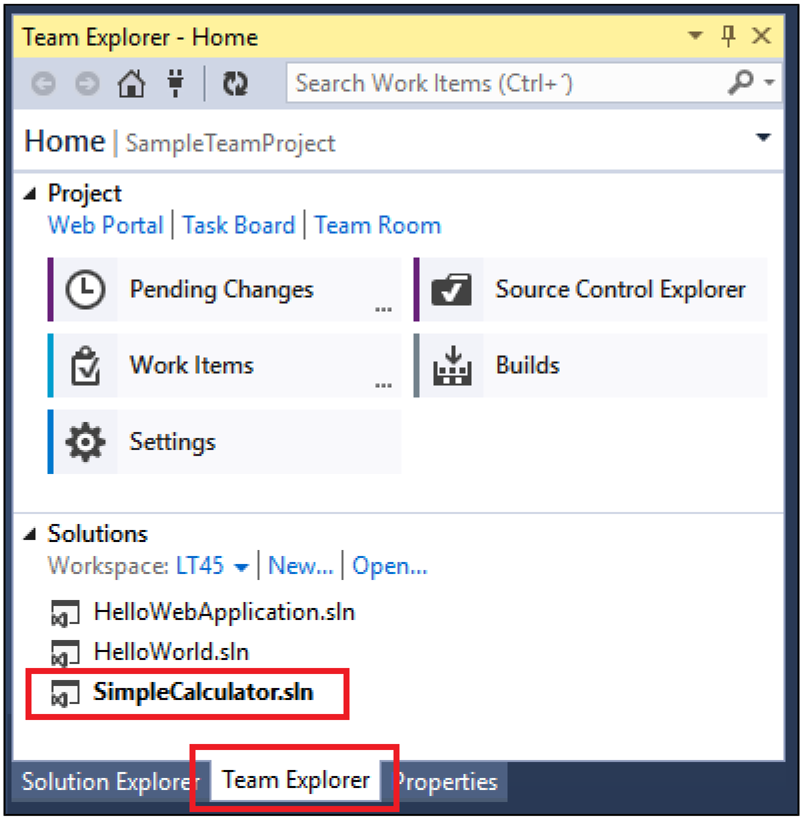
Type a name for the solution folder: SimpleCalculator

Solution and project files will be added to: $/SampleTeamProject/SimpleCalculator

3. Build definition

Once the solution is created in TFS, define a new build process for the solution.

Open the **Team Explorer** tab.



Team Explorer - Home

Search Work Items (Ctrl+´)

Home | SampleTeamProject

Project

Web Portal | Task Board | Team Room

Pending Changes ...

Source Control Explorer

Work Items ...

Builds

Settings

Solutions

Workspace: LT45 | New... | Open...

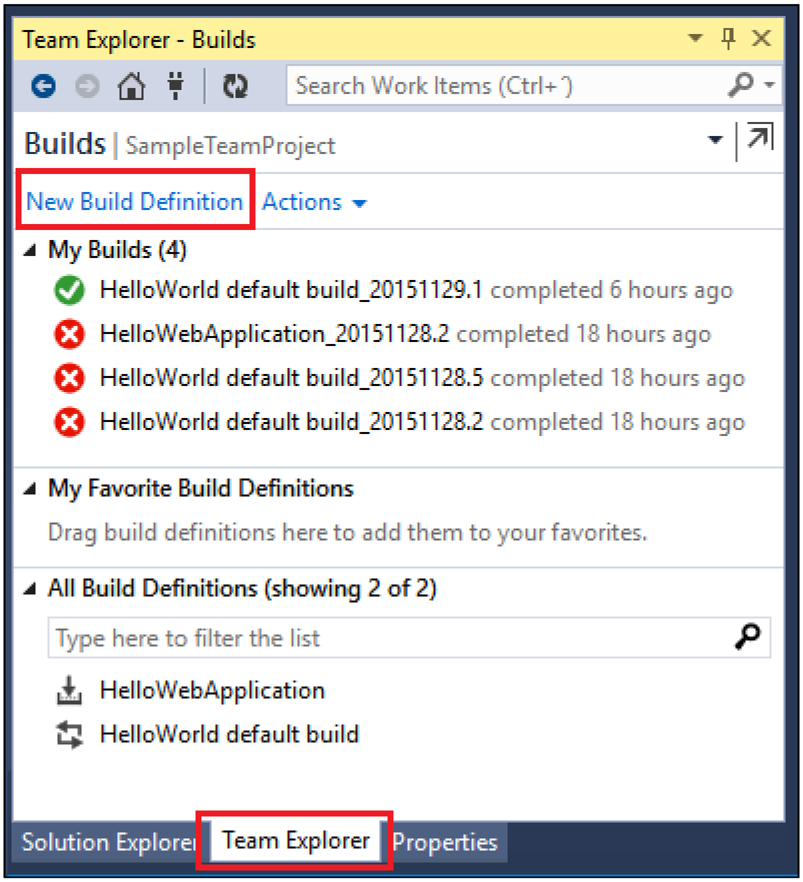
HelloWebApplication.sln

HelloWorld.sln

SimpleCalculator.sln

Solution Explorer | Team Explorer | Properties

Double click on **Builds** and select **New build definition**.



Team Explorer - Builds

Search Work Items (Ctrl+´)

Builds | SampleTeamProject

New Build Definition | Actions

My Builds (4)

- ✓ HelloWorld default build_20151129.1 completed 6 hours ago
- ☒ HelloWebApplication_20151128.2 completed 18 hours ago
- ☒ HelloWorld default build_20151128.5 completed 18 hours ago
- ☒ HelloWorld default build_20151128.2 completed 18 hours ago

My Favorite Build Definitions

Drag build definitions here to add them to your favorites.

All Build Definitions (showing 2 of 2)

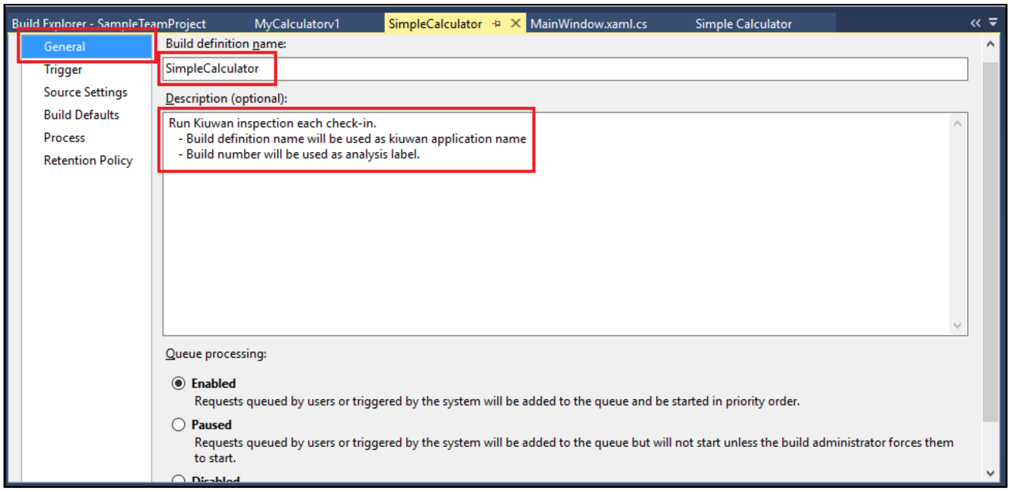
Type here to filter the list

HelloWebApplication

HelloWorld default build

Solution Explorer | Team Explorer | Properties

Below, the images of the dialog boxes to configure the build definition:



Build definition name:

SimpleCalculator

Description (optional):

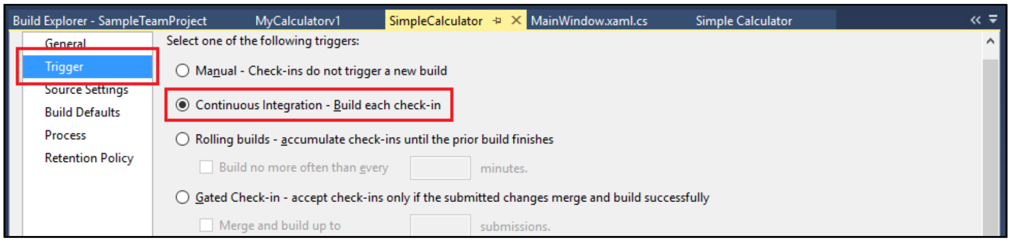
Run Kiuwan inspection each check-in.
 - Build definition name will be used as kiuwan application name
 - Build number will be used as analysis label.

Queue processing:

● **Enabled**
Requests queued by users or triggered by the system will be added to the queue and be started in priority order.

○ **Paused**
Requests queued by users or triggered by the system will be added to the queue but will not start unless the build administrator forces them to start.

○ **Disabled**



Select one of the following triggers:

○ Manual - Check-ins do not trigger a new build

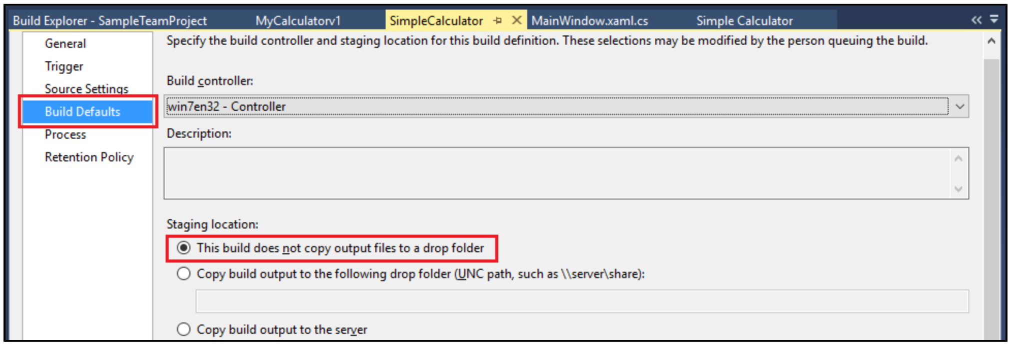
● Continuous Integration - Build each check-in

○ Rolling builds - accumulate check-ins until the prior build finishes

□ Build no more often than every minutes.

○ Gated Check-in - accept check-ins only if the submitted changes merge and build successfully

□ Merge and build up to submissions.



Specify the build controller and staging location for this build definition. These selections may be modified by the person queuing the build.

Build controller:

win7en32 - Controller

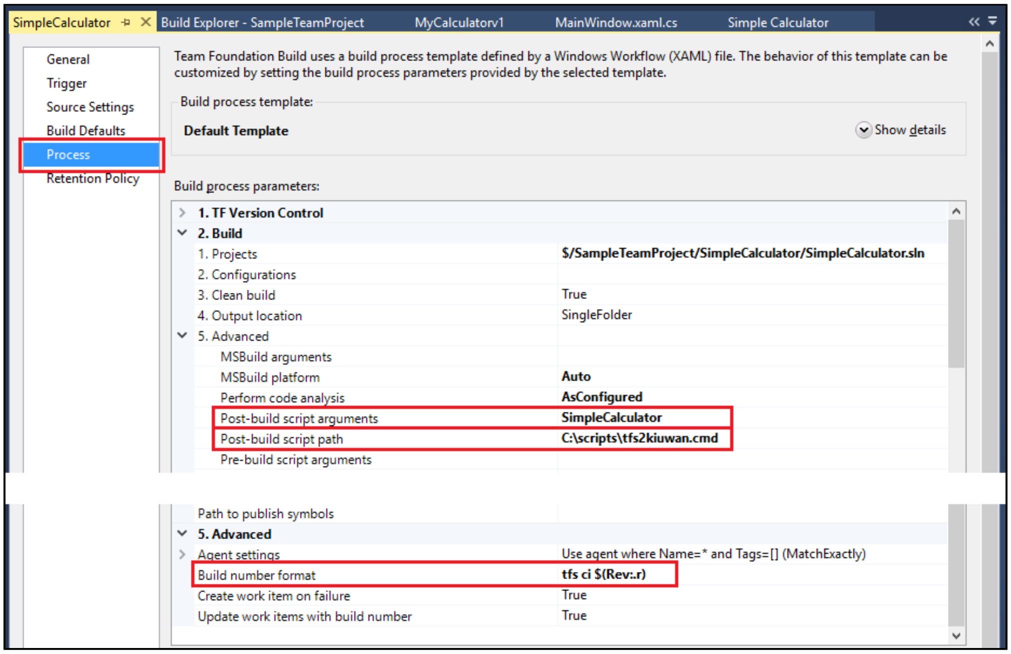
Description:

Staging location:

● This build does not copy output files to a drop folder

○ Copy build output to the following drop folder (UNC path, such as \\server\share):

○ Copy build output to the server



Team Foundation Build uses a build process template defined by a Windows Workflow (XAML) file. The behavior of this template can be customized by setting the build process parameters provided by the selected template.

Build process template:

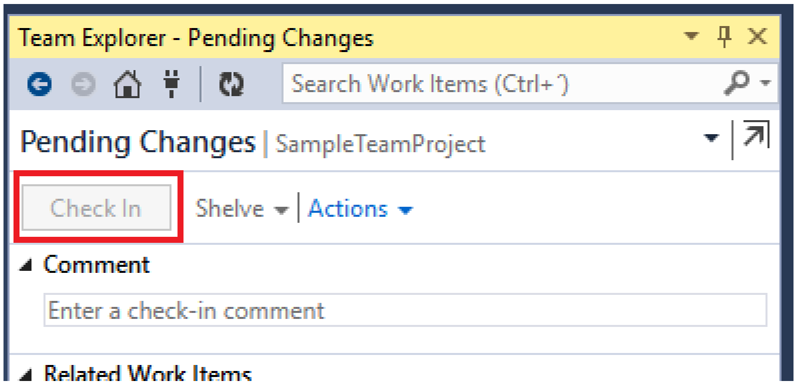
Default Template Show details

Build process parameters:

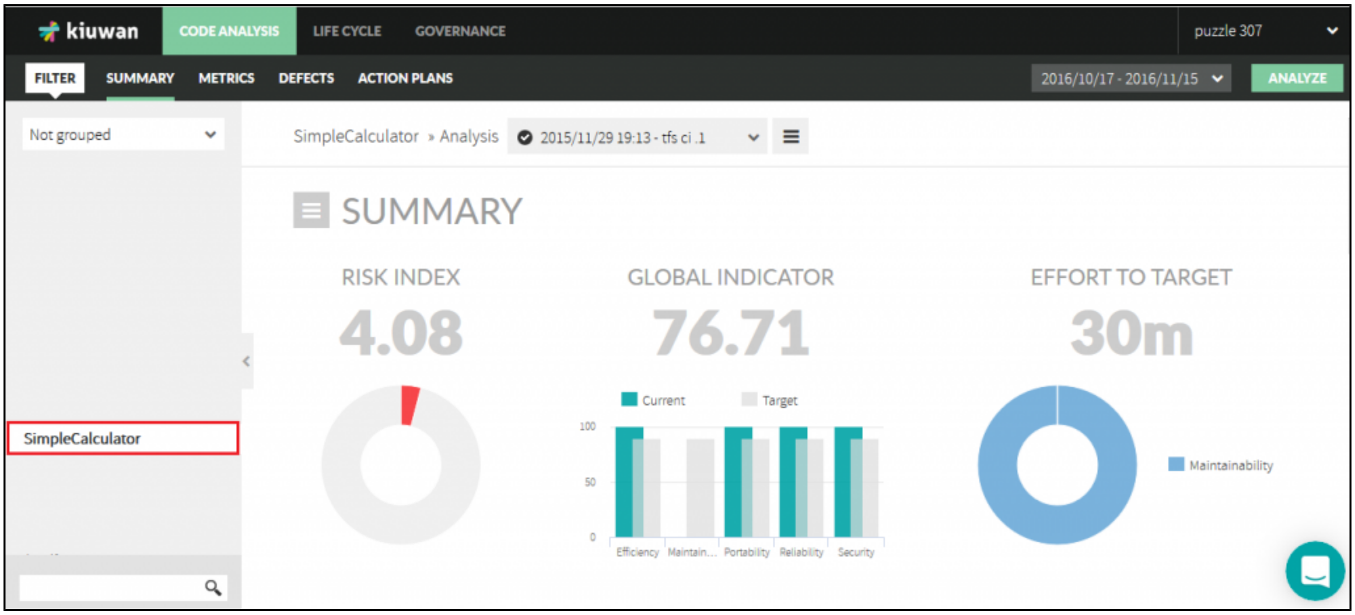
1. TF Version Control	
2. Build	
1. Projects	**$/SampleTeamProject/SimpleCalculator/SimpleCalculator.sln**
2. Configurations	
3. Clean build	True
4. Output location	SingleFolder
5. Advanced	
MSBuild arguments	
MSBuild platform	**Auto**
Perform code analysis	**AsConfigured**
Post-build script arguments	**SimpleCalculator**
Post-build script path	**C:\scripts\tfs2kiuwan.cmd**
Pre-build script arguments	
Path to publish symbols	
5. Advanced	
Agent settings	Use agent where Name=* and Tags=[] (MatchExactly)
Build number format	**tfs ci $(Rev:.r)**
Create work item on failure	True
Update work items with build number	True

4. Commit the changes and get the analysis results in Kiuwan

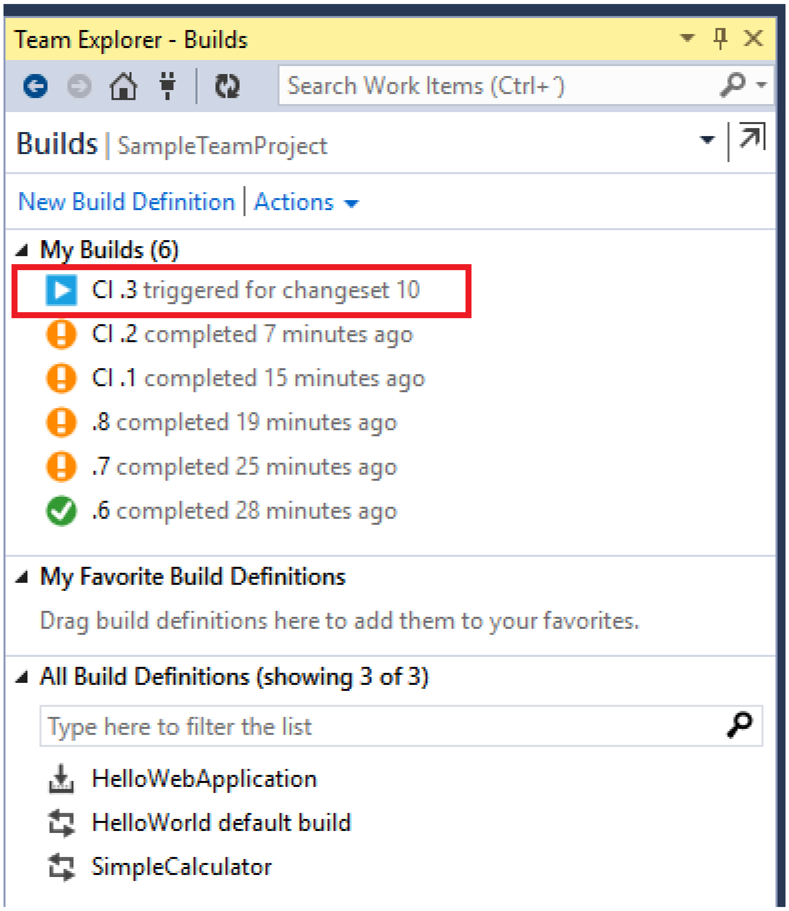
Commit the changes to the repository. A new build will be automatically triggered.


Team Explorer - Pending Changes
Search Work Items (Ctrl+')
Pending Changes | SampleTeamProject
Check In
Shelve
Actions
Comment
Enter a check-in comment
Related Work Items

Login to kiuwan.com to see the results for your application analysis.



For each commit in the repository, a new build and analysis will be run automatically:



Team Explorer - Builds

Search Work Items (Ctrl+')

Builds | SampleTeamProject

New Build Definition | Actions

My Builds (6)

- CI .3 triggered for changeset 10
- CI .2 completed 7 minutes ago
- CI .1 completed 15 minutes ago
- .8 completed 19 minutes ago
- .7 completed 25 minutes ago
- .6 completed 28 minutes ago

My Favorite Build Definitions

Drag build definitions here to add them to your favorites.

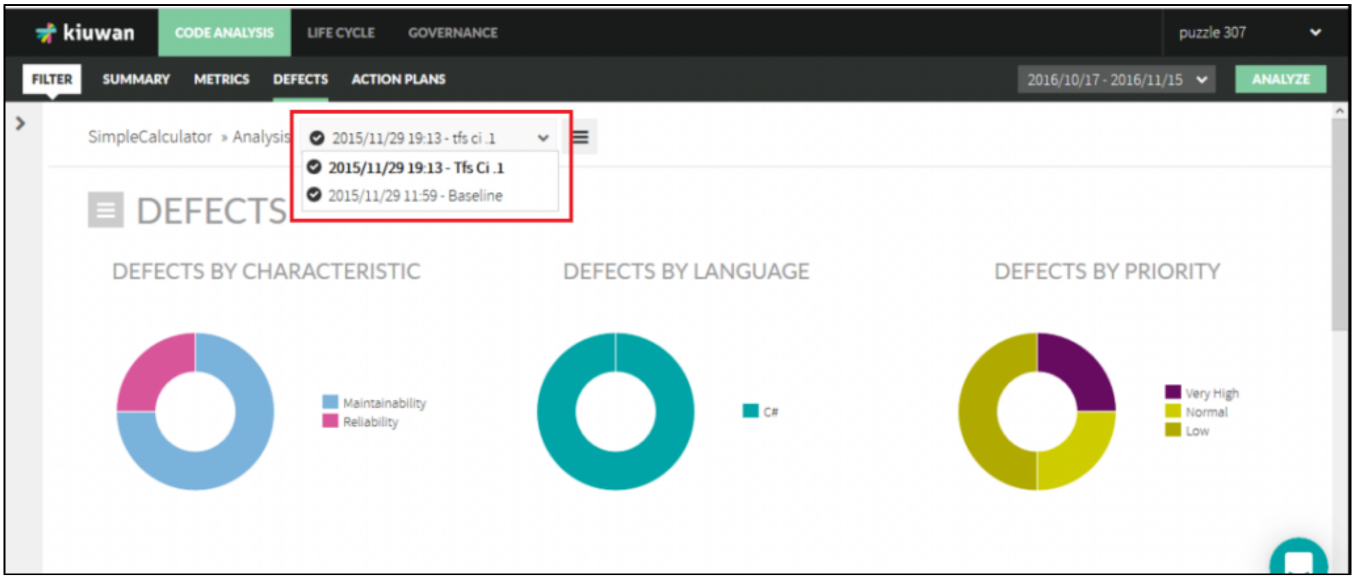
All Build Definitions (showing 3 of 3)

Type here to filter the list

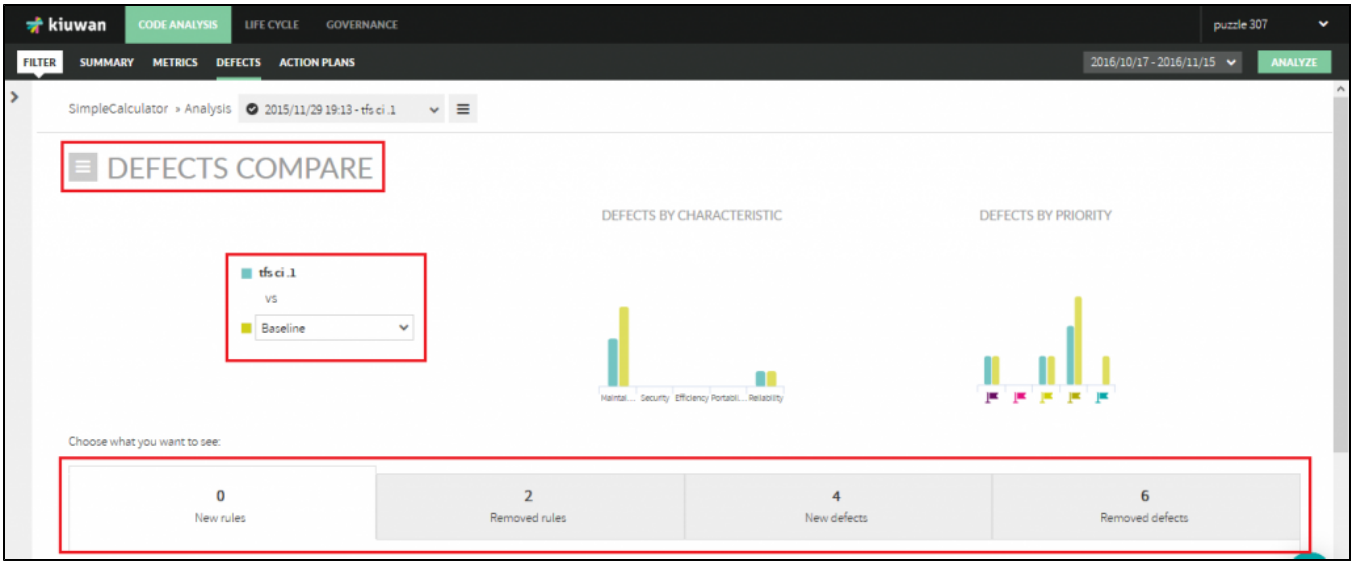
- HelloWebApplication
- HelloWorld default build
- SimpleCalculator

5. Compare each commit against the previous

Each analysis will generate a new version in Kiuwan. You can see all the builds in the **Analysis** drop-down list:



Click **Compare** to see a defect comparison between both analyses.



SimpleCalculator » Analysis 2015/11/29 19:13 - tfs ci .1

DEFECTS COMPARE

🟦 tfs ci .1

vs

🟨 Baseline

Choose what you want to see:

0 New rules	2 Removed rules	4 New defects	6 Removed defects

Now you have a continuous analysis of all your builds in Kiuwan.