Chapter 1: Exploring Test-Driven Development

Write the tests → Write the code → Run the tests → Refactor
Add a Failing Test

Refactor

Make it Work
Chapter 2: Getting Started with .NET Core
Unpacking dotnet-runtime-2.8.0 (2.8.0-1)...
Selecting previously unselected package dotnet-sdk-2.0.2.
Preparing to unpack .../dotnet-sdk-2.0.2_2.0.2-1_amd64.deb ...
Unpacking dotnet-sdk-2.0.2 (2.0.2-1)...
Setting up dotnet-host (2.0.0-1) ...
Setting up aspnetcore-store-2.0.0 (2.0.0-1) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up dotnet-hostfxr-2.0.0 (2.0.0-1) ...
Setting up dotnet-runtime-2.8.0 (2.8.0-1) ...
Setting up dotnet-sdk-2.0.2 (2.0.2-1) ...
This software may collect information about you and your use of the software, and send that to Microsoft.
Please visit http://aka.ms/dotnet-cli-eula for more information.
Welcome to .NET Core!

Learn more about .NET Core @ https://aka.ms/dotnet-docs. Use dotnet --help to see available commands or go to https://aka.ms/dotnet-cli-docs.

.NET Core Tools Telemetry
---------------
The .NET Core tools include a telemetry feature that collects usage information. It is important that the .NET Team understands how the tools are being used so that we can improve them.

The data collected is anonymous and will be published in an aggregated form for use by both Microsoft and community engineers under the Creative Commons Attribution License.

The .NET Core tools telemetry feature is enabled by default. You can opt out of the telemetry feature by setting an environment variable DOTNET_CLI_TELEMETRY_OPTOUT (for example, 'export DOTNET_CLI_TELEMETRY_OPTOUT=1' on macOS/Linux, 'set DOTNET_CLI_TELEMETRY_OPTOUT=1' on Windows) to true (for example, 'true', 1). You can read more about .NET Core tools telemetry at https://aka.ms/dotnet-cli-telemetry.

Installation Note
A command will be run during the install process that will improve project restore speed and enable offline access. It will take a minute to complete.

ayobami@ayobami-VirtualBox:~/Documents/testapp

ayobami@ayobami-VirtualBox:~/Documents/testapp
ayobami@ayobami-VirtualBox:~/Documents/testapp
Welcome to .NET Core!

Learn more about .NET Core @ https://aka.ms/dotnet-docs. Use dotnet --help to see available commands or go to https://aka.ms/dotnet-cli-docs.

Telemetry
---------
The .NET Core tools collect usage data in order to improve your experience. The data is anonymous and does not include command-line arguments. The data is collected by Microsoft and shared with the community.
You can opt out of telemetry by setting a DOTNET_CLI_TELEMETRY_OPTOUT environment variable to 1 using your favorite shell.
You can read more about .NET Core tools telemetry @ https://aka.ms/dotnet-cli-telemetry.

Getting ready...
The template "Console Application" was created successfully.
Processing post-creation actions...

Running 'dotnet restore' on /home/ayobami/Documents/testapp/testapp.csproj...
Restoring packages for /home/ayobami/Documents/testapp/testapp.csproj...
Generating MSBuild file /home/ayobami/Documents/testapp/obj/testapp.csproj.nuget.g.props.
Generating MSBuild file /home/ayobami/Documents/testapp/obj/testapp.csproj.nuget.g.targets.
Restore completed in 306.02 ms for /home/ayobami/Documents/testapp/testapp.csproj.

Restore succeeded.
ayobami@ayobami-VirtualBox:~/Documents/testapp
Updating C# dependencies...
Platform: linux, x86_64, name=ubuntu, version=17.04

Downloading package 'Omnisharp for Linux (x64)' (24595 KB) .................... Done!
Downloading package '.NET Core Debugger (linux / x64)' (54958 KB) .................... Done!

Installing package 'Omnisharp for Linux (x64)'
Installing package '.NET Core Debugger (linux / x64)'

Finished

ayobami@ayobami-VirtualBox:~$  

```
using System;

namespace testapp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Loaded '/usr/share/dotnet/shared/Microsoft.NETCore.App/2.0.0/System.Runtime.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.

Loaded '/usr/share/dotnet/shared/Microsoft.NETCore.App/2.0.0/System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.

Hello World!
The program '[2603] testapp.dll' has exited with code 0 (0x0).
using System;
namespace testapp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}

Hello World!
The program '[2083] testapp.dll' has exited with code 0 (0x0).

Required assets to build and debug are missing from LoanApp! Add them?

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.DependencyInjection.Extensions;

namespace LoanApp
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }
    }
}
Chapter 3: Writing Testable Code

No images
Chapter 4: .NET Core Unit Testing

```csharp
public void Test_calculateLoan_ShouldReturnCorrectRate(Loan loan)
{
    Loan loan = carloanCalculator.calculateLoan(loanDto);
    Assert.IsNotNull(loan);
    Assert.AreEqual(8, loan.InterestRate);
}
```

Extensions.PalindromeCheckerTest.TestCheckWordPalindrome_ShouldReturnTrue

Source: PalindromeCheckerTest.cs line 14

Test Passed - Extensions.PalindromeCheckerTest.TestCheckWordPalindrome_ShouldReturnTrue(word: "civic")
Elapsed time: 0:00:00.059

Test Passed - Extensions.PalindromeCheckerTest.TestCheckWordPalindrome_ShouldReturnTrue(word: "dad")
Elapsed time: 0:00:00.001

Test Passed - Extensions.PalindromeCheckerTest.TestCheckWordPalindrome_ShouldReturnTrue(word: "omo")
Elapsed time: 0:00:00.001
Extensions.PalindromeCheckerTest.Test_IsWordPalindrome_ShouldReturnTrue

Source: PalindromeCheckerTest.cs line 14

- Test Passed - Extensions.PalindromeCheckerTest.Test_IsWordPalindrome_ShouldReturnTrue(word: "civic")
  Elapsed time: 0:00:00.059

- Test Passed - Extensions.PalindromeCheckerTest.Test_IsWordPalindrome_ShouldReturnTrue(word: "dad")
  Elapsed time: 0:00:00.001

- Test Passed - Extensions.PalindromeCheckerTest.Test_IsWordPalindrome_ShouldReturnTrue(word: "omo")
  Elapsed time: 0:00:00.001

Test Explorer

Failed Tests (2)
- LoanApplication.Tests.Unit.GCDTest.Test_GetGcd_ShouldReturnTrue(firstNumber: 30, secondNumber: 20, gcnd: 5) 20 ms
- LoanApplication.Tests.Unit.GCDTest.Test_GetGcd_ShouldReturnTrue(firstNumber: 20, secondNumber: 20, gcnd: 2) 1 ms

Passed Tests (1)
- LoanApplication.Tests.Unit.GCDTest.Test_GetGcd_ShouldReturnTrue(firstNumber: 8, secondNumber: 12, gcnd: 4) 5 ms

Summary
Last Test Run Failed (Total Run Time 0:00:02.2899969)
- 2 Tests Failed
- 1 Test Passed
Chapter 5: Data-Driven Unit Tests
C:\Users\[^\w]+\LoanApplication\MsTest>dotnet test
Build started, please wait...
Build completed.

Test run for C:\Users\[^\w]+\LoanApplication\MsTest\bin\Debug\netcoreapp2.0\MsTest.dll(.NETCoreApp,Version=v2.0)
Microsoft (R) Test Execution Command Line Tool Version 15.3.0-preview-20170628-02
Copyright (c) Microsoft Corporation. All rights reserved.
Starting test execution, please wait...
Total tests: 1. Passed: 1. Failed: 0. Skipped: 0.
Test Run Successful.
Test execution time: 1.2508 Seconds

Extensions and Updates

Installed
  ▲ Online
    ▲ Visual Studio Marketplace
      ▼ Search Results
      ▼ Controls
      ▼ Templates
      ▼ Tools

  ▼ Updates (1)

  ▼ Roaming Extension Manager

Sort by: Relevance

Created by: NUnit Developers
Version: 3.9.0.0
Downloads: 538220
Ratings: ★★★★★ (11 Votes)
More Information
Report Extension to Microsoft

NUnit 3 Test Adapter
NUnit 3 adapter for running tests in Visual Studio. Works with NUnit 3x.

NUnit 2 Test Adapter
NUnit 2 adapter for running tests in Visual Studio 2012 and newer. Works with NUnit 2.x, for 3.x tests use the NUnit 3 adapter.

NUnit VS Templates
Provides Visual Studio project and item templates for NUnit 3 along with code snippets.

Test Generator NUnit extension
Test Generator. NUnit extensions for Visual Studio. Creates Unit tests and Intellisense with both NUnit 2.6.4 and NUnit 3...

NUnit Migrator
Helps transform NUnit v2 to NUnit v3. It consists of a set of Roslyn analyzers and code fixers.

Unit Test Boilerplate Generator
Generates a unit test boilerplate from a given C# class, setting up mocks for all dependencies. Right click an item in Solution Explorer...

SwitchToNUnit3

Change your Extensions and Updates settings

Close
Microsoft.NET.Test.Sdk by Microsoft
The MSbuild targets and properties for building the .Net core test projects.

Microsoft.NETCore.App by Microsoft
A set of .NET API's that are included in the default .NET Core application model.
e8b8861ac7fa042c87a5c2f9f2d04c98b69f28d

NUnit by Charlie Poole, Rob Prouse
NUnit is a unit-testing framework for all .NET languages with a strong TDD focus.

NUnit.ConsoleRunner by Charlie Poole, Rob Prouse
Console runner for the NUnit 3 unit-testing framework, without any extensions.

NUnit3TestAdapter by Charlie Poole, Terje Sandstrom
NUnit 3 adapter for running tests in Visual Studio. Works with NUnit 3.x, use the NUnit 2 adapter for 2.x tests.

Test Explorer
Passed Tests (1)
Test_GetLoanById_ShouldReturnLoan 12 ms
Source: LoanRepositoryTest.cs line 20
Test Passed - Test_GetLoanById_ShouldReturnLoan
Elapsed time: 0:00:00.012
Options

Search Options (Ctrl-E)

- Projects and Solutions
- Source Control
- Work Items
- Text Editor
- Debugging
- IntelliJ Trace
- Performance Tools
- Container Tools
- Cross Platform
- Database Tools
- FF Tools
- Live Unit Testing

General

- Pause Live Unit Testing during build and debug
- Pause Live Unit Testing when battery is below: 30%

Active Solution

- Start Live Unit Testing on solution load

Persisted data directory: D:\Books\TDD C#\LoanApplication\vs\LoanApplic ...

Delete Persisted Data

Test

- Testcase Timeout: 30000 ms
- Maximum number of test processes: 1

Memory Consumption

- Cap memory for Live Unit Testing process at approximately 0 MB

Logging

- Logging Level: Info (Default)

OK Cancel

Test Analyze Window Help

Run
Debug
Playlist
Live Unit Testing
Test Settings
Analyze Code Coverage
Profile Test
Windows

Start Options...
namespace LoanApplication.Controllers
{
    public class HomeController : Controller
    {
        private ILoanRepository loanRepository;

        public HomeController(ILoanRepository loanRepository)
        {
            this.loanRepository = loanRepository;
        }

        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewData["Message"] = "Your application description page.";
            return View();
        }

        public ActionResult Contact()
        {
            ViewData["Message"] = "Your contact page.";
            return View();
        }

        public ActionResult Error()
        {
        }
    }
}
HomeControllerTest+AboutMethod passed
Chapter 6: Mocking Dependencies

Test_GetOlderCarLoanDefaulters_ShouldReturnList failed

Moq.MockException:
Expected invocation on the mock at least 2 times, but was 1 times: x => x.GetCarLoanDefaulters((It.IsInRange<Int32>(1, 12, Range.Inclusive))

Configured setups:
x => x.GetCarLoanDefaulters(It.IsInRange<Int32>(1, 12, Range.Inclusive))

Performed invocations:
ILoanRepository.GetCarLoanDefaulters(12)
at Moq.Mock.ThrowVerifyException(MethodCall expected, IEnumerable`1 setups, IEnumerable`1 actualCalls, Expression expression, Times times, Int32 callCount)
at Moq.Mock.VerifyCalls(Mock targetMock, MethodCall expected, Expression expression, Times times)
at Moq.Mock.Verify[T, TResult](Mock`1 mock, Expression`1 expression, Times times, String failMessage)
at Moq.Mock`1.Verify[TResult](Expression`1 expression, Times times)

Test_GetOlderCarLoanDefaulters_ShouldReturnList failed

Moq.MockException:
Expected invocation on the mock exactly 2 times, but was 1 times: x => x.GetCarloanDefaulters((It.IsInRange<Int32>(1, 12, Range.Inclusive))

Configured setups:
x => x.GetCarLoanDefaulters(It.IsInRange<Int32>(1, 12, Range.Inclusive))

Performed invocations:
ILoanRepository.GetCarLoanDefaulters(12)
at Moq.Mock.ThrowVerifyException(MethodCall expected, IEnumerable`1 setups, IEnumerable`1 actualCalls, Expression expression, Times times, Int32 callCount)
at Moq.Mock.VerifyCalls(Mock targetMock, MethodCall expected, Expression expression, Times times)
at Moq.Mock.Verify[T, TResult](Mock`1 mock, Expression`1 expression, Times times, String failMessage)
at Moq.Mock`1.Verify[TResult](Expression`1 expression, Times times)
using Moq;
using Xunit;

namespace LoanApplication.Tests.Unit
{
    public class LoanServiceTest
    {
        private Mock<ILoanRepository> loanRepository;
        private LoanService loanService;

        public LoanServiceTest()
        {
            loanRepository.Setup(x => x.GetCanLoanDefaulters(1, 12)).Returns(() => new[] { new Person { FirstName = "Donald", LastName = "Duke", Age = 30 }, new Person { FirstName = "Ayobami", LastName = "Adewole", Age = 20 } });
            loanRepository.Setup(x => x.GetLoanDefaultersInRange(1, 12, Range.Inclusive)).Returns(() => new[] { new Person { FirstName = "Donald", LastName = "Duke", Age = 30 }, new Person { FirstName = "Ayobami", LastName = "Adewole", Age = 20 } });
            loanService = new LoanService(loanRepository.Object);
        }
    }
}
Chapter 7: Continuous Integration and Project Hosting

Learn Git and GitHub without any code!
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide  Start a project
Create a new repository

A repository contains all the files for your project, including the revision history.

Owner  Repository name
ayobami  LoanApplication

Great repository names are short and memorable. Need inspiration? How about vigilant-invention.

Description (optional)
TDD in C# 7 Book

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you’re importing an existing repository.

Add .gitignore: None  Add a license: None

Create repository
TDD in C# 7 Book

Add topics

- 1 commit
- 2 branches

Branch: Loan-Rate-Fix

New pull request

This branch is even with master.

- ayobami Initial commit

- README.md Initial commit

- README.md

ayobami Updated the readme file

Showing 1 changed file with 3 additions and 0 deletions.

<table>
<thead>
<tr>
<th>▼ ▼ ▼ ▼ README.md</th>
</tr>
</thead>
<tbody>
<tr>
<td>... ...</td>
</tr>
<tr>
<td>@@ -1,2 +1,5 @@</td>
</tr>
<tr>
<td>1 1 # loanApplication</td>
</tr>
<tr>
<td>2 2 TDD in C# 7 Book</td>
</tr>
<tr>
<td>3 +</td>
</tr>
<tr>
<td>4 +</td>
</tr>
<tr>
<td>5 +</td>
</tr>
<tr>
<td>Made Changes to this file, in order to make a pull request</td>
</tr>
</tbody>
</table>
Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.
Note: GitHub Services are being deprecated. Please contact your integrator for more information on how to migrate or replace a service to webhooks or GitHub Apps.

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you’d like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *

http://localhost:54113/API/webhook

Content type

application/json

Secret

Which events would you like to trigger this webhook?

- Just the push event.
- Send me everything.
- Let me select individual events.

Active

We will deliver event details when this hook is triggered.

Add webhook
Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we’ll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

Last delivery was not successful. Couldn’t connect to server.

http://localhost:54113/API/webhook (create, issues, and push)

ngrok by @inconshreveable

Session Status          online
Session Expires         7 hours, 55 minutes
Version                 2.2.8
Region                  United States (us)
Web Interface           http://127.0.0.1:4040
Forwarding              http://d73c1ef5.ngrok.io -> localhost:54113
                        https://d73c1ef5.ngrok.io -> localhost:54113
Connections
                        ttl   opn   rt1   rt5   p50   p90
                        3      0     0.01  0.01  120.71 122.43
HTTP Requests

POST /api/webhook 200 OK
POST /api/webhook 200 OK
GET /favicon.ico   200 OK
GET /api/webhook   200 OK
Recent Deliveries

217bf560-1f40-11e8-8d7d-20e964da8f46

Request Response 200

Completed in 0 seconds.

Headers

Request URL: http://d73c1ef5.ngrok.io/api/webhook
Request method: POST
content-type: application/json
Expect:
User-Agent: GitHub-Hookshot/7a71d82
X-GitHub-Delivery: 217bf560-1f40-11e8-8d7d-20e964da8f46
X-GitHub-Event: ping
TeamCity First Start

Please review the settings below before proceeding with the first TeamCity start.

TeamCity server stores server configuration settings, project definitions, build results and caches on disk in a **Data Directory**.

Data Directory location on the TeamCity server machine:

```
C:\ProgramData\JetBrains\TeamCity
```

If you already worked with TeamCity on this machine you can specify existing Data Directory.

Proceed
Database connection setup

TeamCity server stores builds history and users-related data in an SQL database.

Select the database type: Internal (HSQLDB)

The internal database suits evaluation purposes only and is not intended for production. We strongly recommend using an external database in a production environment.

You can start with the internal database and then migrate the data to an external one after successful evaluation.

Proceed

TeamCity 2017.2.2 (build 50909)
Getting started with TeamCity

There are no projects in TeamCity. To start running builds, create projects and build configurations first.

+ Create project

You may also want to:

- configure email and Jabber settings to enable notifications,
- manage licenses, and
- add more users to TeamCity.

Create Project

- From a repository URL
  - From GitHub.com
  - From Bitbucket Cloud
    - Set up connection
- From Visual Studio Team Services
  - Set up connection
- Manually
Register a new OAuth application

Application name
LoanApplication

Something users will recognize and trust

Homepage URL
http://localhost:8060

The full URL to your application homepage

Application description
Application description is optional

This is displayed to all users of your application

Authorization callback URL
http://localhost:8060/oauth/github/accessToken.html

Your application's callback URL. Read our OAuth documentation for more information.

[Register application] [Cancel]
Please register TeamCity on GitHub using the following parameters:
Homepage URL: http://localhost:8060
Callback URL: http://localhost:8060/oauth/github/accessToken.html

And once registered, enter the client id and the client secret in the form below.

Display name:* GitHub.com
Provide some name to distinguish this connection from others.

Client ID:* 345200000000130

Client secret:* 

Choose a repository:* 

Your repositories (5)
7WCM0033 (https://github.com/ayobami/7WCM0033)
ChildrenLearning (https://github.com/ayobami/ChildrenLearning)
Currency-Converter (https://github.com/ayobami/Currency-Converter)
LoanApplication (https://github.com/ayobami/LoanApplication)
VCS repository connection has been verified. Please review project and build configuration names and click **Proceed** to create.

- **Project name**: LoanApplication
- **Build configuration name**: Build
- **VCS Repository**: (Git) https://github.com/ayobami/LoanApplication

---

**New Build Step**

- **Runner type**: .NET CLI (dotnet)
  - Provides .NET CLI toolchain support for .NET projects
- **Step name**: Run Unit Tests
- **Command**: test
- **Projects**: Specify paths to projects and solutions. Wildcards are supported.

---

**Build Steps**

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with

<table>
<thead>
<tr>
<th>Build Step</th>
<th>Parameters Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Run Unit Tests</td>
<td>.NET CLI (dotnet) test * Execute if all previous steps finished successfully</td>
</tr>
</tbody>
</table>
#1 (06 Mar 18 13:54)

<table>
<thead>
<tr>
<th>Status</th>
<th>Running</th>
</tr>
</thead>
<tbody>
<tr>
<td>Progress</td>
<td>3s passed</td>
</tr>
<tr>
<td>Thread dump</td>
<td>View thread dump</td>
</tr>
</tbody>
</table>
Chapter 8: Creating Continuous Integration
Build Processes

---

```bash
Administrator: Windows PowerShell

NUnit Console Runner 3.4.0
Copyright (C) 2016 Charlie Poole

Runtime Environment
  OS Version: Microsoft Windows NT 10.0.15063.0
  CLR Version: 4.0.30319.42000

Test Files
  C:/Users/ExampleTests/bin/Release/ExampleTests.dll

Run Settings
  Working Directory: C:/Users/Example
  ImageRuntimeVersion: 4.0.30319
  ImageTargetFrameworkName: .NETFramework,Version=v4.5
  ImageRequiresX86: False
  ImageRequiresDefaultAppDomainAssemblyResolver: False
  NumberOfTestWorkers: 4

Test Run Summary
  Overall result: Passed
  Test Count: 2, Passed: 2, Failed: 0, Inconclusive: 0, Skipped: 0
  Duration: 0.113 seconds

---

Default

Task                              Duration
------------------------------------
Clean                              00:00:00.0161013
Restore-NuGet-Packages            00:00:31.8511678
Build                              00:00:03.6143534
Run-Unit-Tests                    00:00:02.9733688
------------------------------------
Total                              00:00:36.4855784

PS C:/Users/Example

---

Windows PowerShell

PS C:/Users/Example/Downloads/Example-master> .\build.ps1
.\uild.ps1 : File C:/Users/Example/Downloads/Example-master/build.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at
https://go.microsoft.com/fwlink/?linkid=135178.
At line:1 char:1
  + .\build.ps1
  + CategoryInfo : SecurityError: (:) [], PSSecurityException
  + FullyQualifiedErrorId : UnauthorizedAccess
PS C:/Users/Example/Downloads/Example-master>
```
### Get-ExecutionPolicy -List

```
Scope            ExecutionPolicy
----------------- ---------------------
MachinePolicy    Undefined
UserPolicy       Undefined
Process          Undefined
CurrentUser      Undefined
LocalMachine     Undefined
```

### Set-ExecutionPolicy Remotesigned -Scope Process

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about Execution_Policies help topic at [https://go.microsoft.com/fwlink/?linkid=135170](https://go.microsoft.com/fwlink/?linkid=135170). Do you want to change the execution policy?
[Y]: [N] No to All [S] Suspend [?] Help (default is "N"): Y

<table>
<thead>
<tr>
<th>Configuration option</th>
<th>Description</th>
<th>Default value</th>
<th>Environment variable name</th>
<th>Config file content</th>
<th>Direct argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools path</td>
<td>This option is used to tell Cake which folder contains the tools to be used by Cake when restoring.</td>
<td>.\tools</td>
<td>CAKE_PATHS_TOOLS</td>
<td>[Paths]tools=./tools</td>
<td>cake.exe --paths_tools=./tools</td>
</tr>
<tr>
<td>Addins path</td>
<td>This config option is used to specify the Addins folder to be used by cake while restoring Add-ins.</td>
<td>.\tools\Addins</td>
<td>CAKE_PATHS_ADDINS</td>
<td>[Paths]Addins=./tools/Addins</td>
<td>cake.exe --paths_addins=./tools/Addins</td>
</tr>
<tr>
<td>NuGet download URL</td>
<td>Using this option, you can specify where Cake should download NuGet packages from when you are using the addin and tool preprocessor.</td>
<td><a href="https://packages.nuget.org/api/v2">https://packages.nuget.org/api/v2</a></td>
<td>CAKE_NUGET_SOURCE</td>
<td>[NuGet]Source= <a href="http://someurl/nuget">http://someurl/nuget</a></td>
<td>cake.exe --nuget_source=<a href="http://someurl/nuget/">http://someurl/nuget/</a></td>
</tr>
<tr>
<td>Modules path</td>
<td>This is used for configuring the Modules folder for use by Cake when loading custom modules.</td>
<td>.\tools\Modules</td>
<td>CAKE_PATHS_MODULES</td>
<td>[Paths]modules=./tools/M helpless</td>
<td>cake.exe --paths_modules=./tools/Modules</td>
</tr>
<tr>
<td>Skip verification</td>
<td>This option is used to avoid runtime errors that can occur during script execution. When exceptions occur during script execution, Cake will abort the execution of the script.</td>
<td>false</td>
<td>CAKE_SETTINGS_SKIPVERIFICATION</td>
<td>[Settings]SkipVerification=true</td>
<td>cake.exe --settings_skiplification=true</td>
</tr>
<tr>
<td>Add-in NuGet dependencies</td>
<td>A configuration option that becomes available since Cake Version 0.2.0 to install and reference NuGet package dependencies.</td>
<td>false</td>
<td>CAKE_NUGET_LOADDEPENDENCIES</td>
<td>[NuGet]loadDependencies=true</td>
<td>cake.exe --nuget_loaddependencies=true</td>
</tr>
</tbody>
</table>
Preparing to run build script...
Running build script...

----------------------------------------
Clean
----------------------------------------

----------------------------------------
Restore-NuGet-Packages
----------------------------------------
Restoring NuGet Packages

Welcome to .NET Core!

Learn more about .NET Core @ https://aka.ms/dotnet-docs. Use dotnet --help to see available commands or go to https://aka.ms/dotnet-cli-docs.

Telemetry
----------
The .NET Core tools collect usage data in order to improve your experience. The data is anonymous and does not include command-line arguments. The data is collected by Microsoft and shared with the community.
You can opt out of telemetry by setting a DOTNET_CLI_TELEMETRY_OPTOUT environment variable to 1 using your favorite shell.

0 Error(s)

Time Elapsed 00:01:07.79

----------------------------------------
Run-Tests
----------------------------------------

----------------------------------------
Default
----------------------------------------

<table>
<thead>
<tr>
<th>Task</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>00:00:00.0248073</td>
</tr>
<tr>
<td>Restore-NuGet-Packages</td>
<td>00:00:29.1532166</td>
</tr>
<tr>
<td>Build</td>
<td>00:01:17.0493209</td>
</tr>
<tr>
<td>Run-Tests</td>
<td>00:00:00.0128710</td>
</tr>
<tr>
<td>Total:</td>
<td>00:01:46.2433470</td>
</tr>
</tbody>
</table>
Install Complete

'Cake for Visual Studio' has been successfully installed. Please close and restart all target application instances for changes to take effect.

- Microsoft Visual Studio Enterprise 2015
- Visual Studio Enterprise 2017

View Install Log  Close
build.cake

```csharp
// Arguments
var target = Argument("target", "Default");
var solution = Argument("configuration", "Release");

// Define directories.
var buildDir = Directory("./LoanApplication.Core/bin") + Directory(configuraiton);

// Tasks
Task("Clean")
  .Does(() =>
    CleanDirectory(buildDir);
  );

Task("Restore-NuGet-Packages")
  .IsDependentOn("Clean")
  .Does(() =>
    Information("Restoring NuGet Packages");
    DotNetCoreRestore();
  );

Task("Build")
  .IsDependentOn("Restore-NuGet-Packages")
  .Does(() =>
    Information("Build Solution");
    DotNetCoreBuild(solution,
      new DotNetCoreBuildSettings()
    );
```

Solution called LoanApplication

C:\Users\[User]\Documents\Visual Studio 2017\Projects\book\LoanApplication> C:\Users\[User]\Documents\Visual Studio 2017\Projects\book\LoanApplication\tools\Cake\Cake.exe
build.cake -Target="Clean"

Clean
---
Task Duration
---
Clean 00:00:00.00392250
Total: 00:00:00.00392250
Process terminated with code 0.
### New Build Step

**Runner type:** PowerShell

**Step name:** Cake Build

**Script:** File

**Script file:** Path to the PowerShell script, relative to the checkout directory

**Script execution mode:** Execute .ps1 from external file

**Script arguments:** Expand

Show advanced options

---

### Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool.

<table>
<thead>
<tr>
<th>Build Step</th>
<th>Parameters Description</th>
</tr>
</thead>
</table>
| 1. Cake Build | PowerShell <Any Binness> File: build.ps1
|             | Execute: If all previous steps finished successfully |
Host my projects at:

ayobami.visualstudio.com

Manage code using:

- Git
- Team Foundation Version Control

We will host your projects in **Central US** location.
You can share work with other users.

[Change details]

[Continue]

To keep our lawyers happy:
By continuing, you agree to the [Terms of Service], [Privacy Statement], and [Code of Conduct].
Create new project

Projects contain your source code, work items, automated builds and more.

Project name *
LoanApplication

Description
LoanApplication

Version control
Git

Work item process
Agile

Create  Cancel
Get started with your new project!

- **Cloning to your computer**
  - HTTPS
  - SSH
  - `https://ayobami.visualstudio.com/DefaultCollection/_git/LoanApplication`
  - OR
  - Generate Git credentials
  - Cloned in Visual Studio

  - Having problems authenticating in Git? Be sure to get the latest version of Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

- **Pushing an existing repository from command line**

- **Importing a repository**

- **Initializing with a README or gitignore**
Import a Git repository

Source type

Git

Clone URL *

https://github.com/ayobami/LoanApplication.git

Requires authorization

Username

ayobami

Password / PAT *

[Blurred]

Import Close
Import successful!

Congratulations! Your https://github.com/ayobami/LoanApplication.git repository has been successfully imported.

If you are not automatically redirected to your repository page [Click here to navigate to code view.](#)
Cake
Cake Build | 1,268 installs | ★★★★★ (8) | Free

Tools for building with Cake.

Get it free

Select a Visual Studio Team Services account

aycbami

Install

Download
Save build definition and queue

Save comment
Configured Cake Script

Agent queue
Hosted VS2017

Branch
master

Commit
Configured Cake Script

**Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BuildConfiguration</td>
<td>release</td>
</tr>
<tr>
<td>BuildPlatform</td>
<td>any cpu</td>
</tr>
<tr>
<td>system.debug</td>
<td>false</td>
</tr>
</tbody>
</table>

**Demands**

**Add**

[Save & queue] [Cancel]
Chapter 9: Testing and Packaging the Application

 Command Prompt

C:\LoanApplication\LoanApplication>dotnet publish
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

LoanApplication.Core -> C:\LoanApplication\LoanApplication.Core\bin\Debug\netcoreapp2.0\LoanApplication.Core.dll
LoanApplication -> C:\LoanApplication\LoanApplication\bin\Debug\netcoreapp2.0\LoanApplication.dll
LoanApplication -> C:\LoanApplication\LoanApplication\bin\Debug\netcoreapp2.0\publish\n
C:\LoanApplication\LoanApplication>

 Command Prompt

C:\LoanApplication\LoanApplication>dotnet pack C:\LoanApplication\LoanApplication.Core\LoanApplication.Core.csproj --output
Microsoft (R) Build Engine version 15.4.8.50001 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

LoanApplication.Core -> C:\LoanApplication\LoanApplication.Core\bin\Debug\netcoreapp2.0\LoanApplication.Core.dll
Successfully created package 'C:\LoanApplication\LoanApplication.Core\nupkgs\LoanApplication.Core.1.0.0.nupkg'.
dotnet nuget push LoanApplication.Core.1.0.0.nupkg -k oy2ako4kjecbduemf3

PUT https://www.nuget.org/api/v2/package/

Created https://www.nuget.org/api/v2/package/ 4882ms

Your package was pushed.

Published Packages

<table>
<thead>
<tr>
<th>Package ID</th>
<th>Owners</th>
<th>Downloads</th>
<th>Latest Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoanApplication.Core</td>
<td>ayobami</td>
<td>0</td>
<td>1.0.0</td>
</tr>
</tbody>
</table>