Exercise 1 “Create a basic console application”

In this exercise you will create a small C# project and will see how changes in the project configuration are translated to their equivalent MSBuild Syntax

After each step save all open files and look at the project file in a text editor.

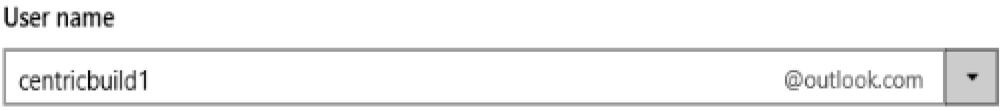
1. Create new Console Application
2. Open the Configuration Manager and setup Any CPU, x64 and x86 build configurations. When asked to create a matching project configuration choose: **yes**
3. Add a readme.txt to the project
4. Investigate the internals of the project file
    a. Unload project
    b. Edit project
5. Run MSBuild X64 from commandline

Exercise 2 “Setting up the accounts”

In this exercise you will create a Microsoft Account. With this account you will create a trail VSTS subscription.

Let’s start by creating a Microsoft Account

6. Start an InPrivate / incognito browser session
7. Grab your numbered post-it from the wall, please stick to this number!
8. Navigate to https://signup.live.com/
9. Complete the registration form.



User name

centricbuild1 @outlook.com

10. Write down your email and password! Save to a file on the desktop
11. Please use a valid alternate email and phone, this allows you to reset the account.

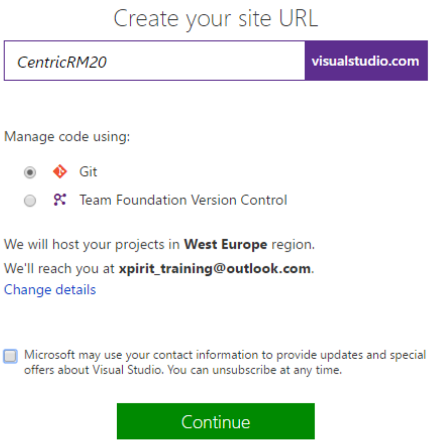
Now we will create a brand new Visual Studio environment

1. Navigate to https://www.visualstudio.com/
2. Choose Visual Studio Team Services “Get started for free”
3. Sign in with the previously create account
4. Create your site URL use format “CentricBuild” + your post-it number

   e.g. CentricBuild1

   Please choose Git for source control

   Accept and Continue



Create your site URL

CentricRM20 visualstudio.com

Manage code using:

● Git

○ Team Foundation Version Control

We will host your projects in **West Europe** region.
We'll reach you at **xpirit_training@outlook.com**.
Change details

□ Microsoft may use your contact information to provide updates and special offers about Visual Studio. You can unsubscribe at any time.

Exercise 3 “Setting up the environment”

In this exercise we will connect to the Azure VM prepared for you. Please use the VM number picked by you from the board. Follow the steps to get started.

1. Start the Remote Desktop Connection tool
2. Use the following address, replace X with your picked number.

   XpiritCentricX.westeurope.cloudapp.azure.com

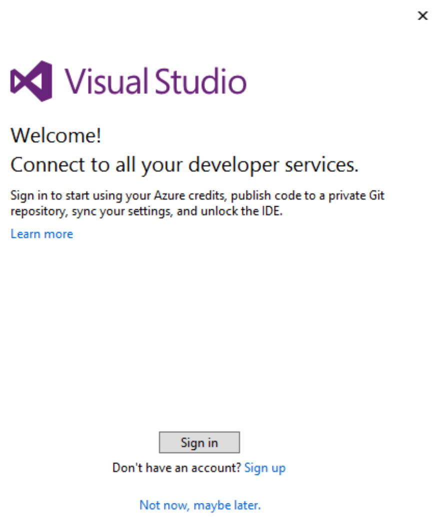
3. Use the following user account: \CentricAdmin
4. Use the following password: **P@ssw0rd1**

Now we will prepare the server

1. Start the Server Manager
2. Navigate to the Local Server
3. From the server properties choose to change “IE Enhance Security Configuration”
4. Set both options to “Off”

Now let’s prepare our development environment

1. Start Visual Studio 2015
2. In the welcome dialog choose to **Sign In**
3. Provide the account details created in exercise 1.
4. Continue the preparation of Visual Studio.
5. Validate if you are signed in to Visual Studio, in the top left corner you should see your account name.
6. If not, please sign in through the account settings


×
Visual Studio
Welcome!
Connect to all your developer services.
Sign in to start using your Azure credits, publish code to a private Git repository, sync your settings, and unlock the IDE.
Learn more
Sign in
Don't have an account? Sign up
Not now, maybe later.

As last step in the preparations we will connect to our created VSTS environment.

1. In Visual Studio, navigate to the Team Explorer
2. Choose [Manage Connections]
3. In the Connect to Team Foundation Server dialog click the [Servers…] button
4. Your VSTS environment should be listed. Select it from the list
5. Rename Project to **CentricBuild**
6. Mark the checkbox for “**CentricBuild**”
7. Then click [Connect]

Your team explorer now connects to this project. Let’s setup our working environment.

1. Choose to clone the repository.
2. Your repository URL should be [https://centricrmX.visualstudio.com/_git/CentricBuild] The X represents the number on your post-it.

3. Accept the suggested working directory [C:\Users\CentricAdmin\Source\Repos\CentricBuild]
4. Now click [Clone] to get your local copy of the repository.
5. Download Solutions.zip from https://github.com/JasperGilhuis/BuildTraining
6. Unpack into root directory C:\Users\CentricAdmin\Source\Repos\CentricBuild
7. Commit the code to VSTS.

Exercise 4 “Create a new website and build”

In this exercise we will create an out-of-the-box web application, which we will add to the Git repository and then we will create a build definition for it.

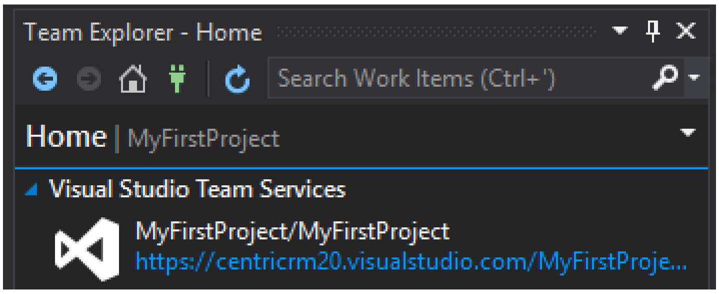
1. Form the Visual Studio menu choose [File, New, Project]
2. Pick [ASP.NET Web Application] (.NET framework 4.5.2)
3. Deselect the [Add Application Insights to project]
4. Provide a name [CentricRMWeb**X]**
5. Choose the correct location [C:\Users\CentricAdmin\Source\Repos\MyFirstProject]
6. Choose [MVC] from the ASP.NET 4.5.2 Templates
7. Deselect the [Host in the cloud] checkbox
8. Click [OK] to get started
9. You should now see the [Your ASP.NET application]

Now let’s add this solution to our Git Repository.

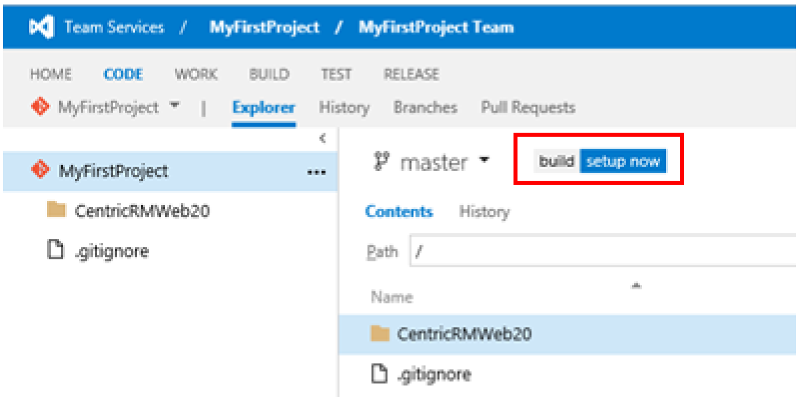
1. Navigate to the Team Explorer
2. Navigate to [Changes]
3. Exclude unwanted items from the changes;
   a. Ignore [.suo] by ignoring the extension
   b. Ignore the [bin] folder
   c. Ignore the [obj\bin] folder
   d. Ignore the [packages] folder
4. Now stage the changes
5. Provide a check-in message [Initial add of application]
6. [Commit and Sync] the stages

Now we will navigate to the VSTS environment.

1. In your Team Explorer a link to your VSTS environment should be available


Team Explorer - Home
Search Work Items (Ctrl+')
Home | MyFirstProject
Visual Studio Team Services
MyFirstProject/MyFirstProject
https://centricrm20.visualstudio.com/MyFirstProje...

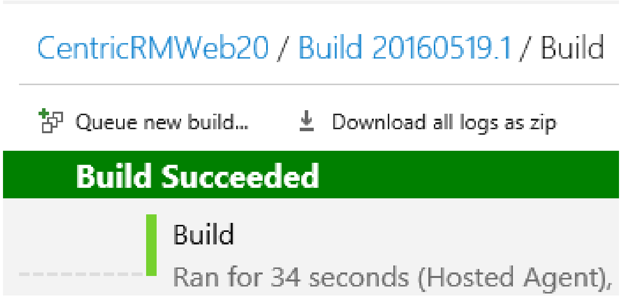
2. You should now see the following screen



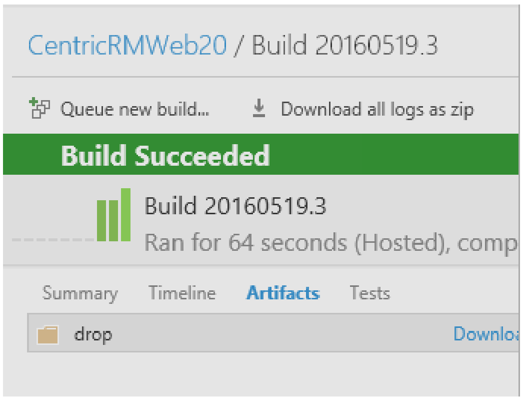
3. Click the [setup now] button to create a build definition for our project.
4. Follow the wizard for [Visual Studio] and click [Next]
5. Your repository details should be provided, choose [Create]
6. Navigate the [Build solution] step, provide the MSBuild Arguments

```
/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true
/p:SkipInvalidConfigurations=true /p:PackageLocation=$(build.stagingDirectory)
```

7. Choose to [Save] the build definition
8. Provide a name [CentricRMWeb**X**]
9. Queue a new build to validate it. The hosted agent should run it without issues. The hosted agent queue may take a while to start.


CentricRMWeb20 / Build 20160519.1 / Build
Queue new build...
Download all logs as zip
Build Succeeded
Build
Ran for 34 seconds (Hosted Agent),

10. Validate the if your build has artifacts



CentricRMWeb20 / Build 20160519.3

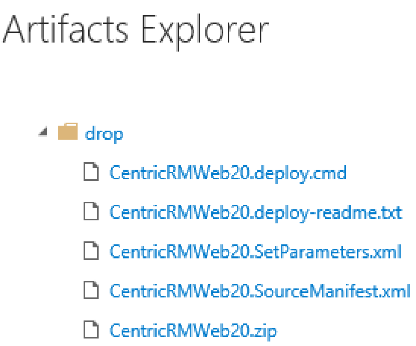
Queue new build... Download all logs as zip

Build Succeeded

Build 20160519.3
Ran for 64 seconds (Hosted), comp

Summary Timeline **Artifacts** Tests

drop Downloa



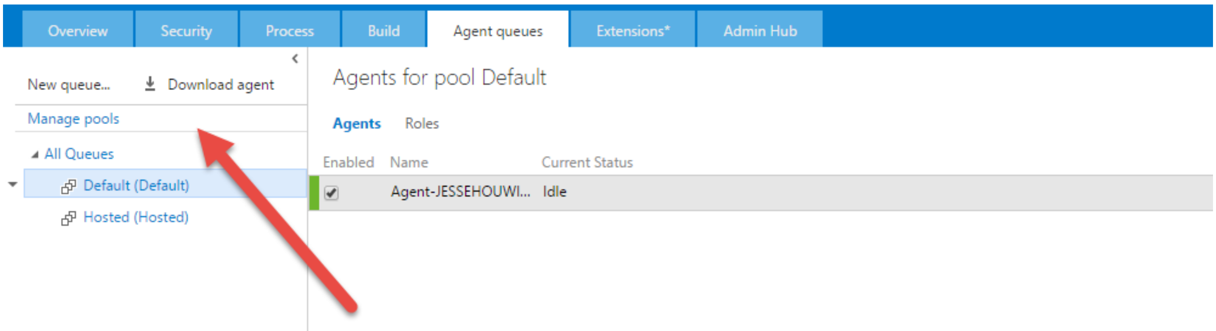
Artifacts Explorer

- drop
  - CentricRMWeb20.deploy.cmd
  - CentricRMWeb20.deploy-readme.txt
  - CentricRMWeb20.SetParameters.xml
  - CentricRMWeb20.SourceManifest.xml
  - CentricRMWeb20.zip

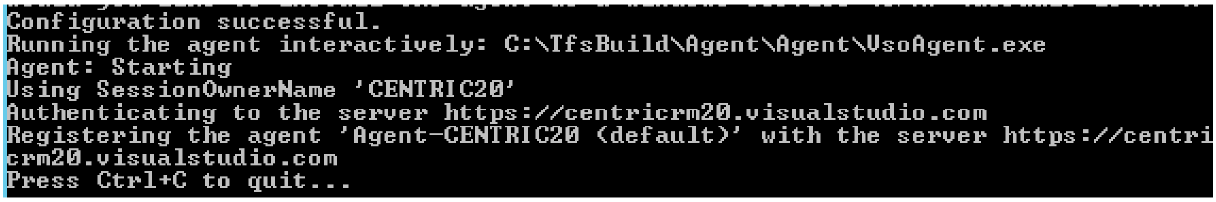
Exercise 5 “Setting up a local build agent”

In this exercise you will setup a local agent to simplify debugging and improve your build’s performance.

1. Navigate to https://centricrmX.visualstudio.com/_admin
2. Download the agent from VSTS



3. Extract the agent to a local folder (e.g. c:\TfsBuild\Agent)
4. Run the ConfigureAgent.cmd from an administrator command prompt
    - Accept the default name
    - Use your Visual Studio account Uri (https://centricrmX.visualstudio.com/)
    - Use the default pool
    - Accept default working folder
    - Sign in to your visual studio account with previously created credentials
    - Choose not to run the agent as a windows service
    - Validate your agent is running



```
Configuration successful.
Running the agent interactively: C:\TfsBuild\Agent\Agent\VsoAgent.exe
Agent: Starting
Using SessionOwnerName 'CENTRIC20'
Authenticating to the server https://centricrm20.visualstudio.com
Registering the agent 'Agent-CENTRIC20 (default)' with the server https://centri
crm20.visualstudio.com
Press Ctrl+C to quit...
```

5. Keep this Command Prompt running.
6. Validate in VSTS that your agent is registered.

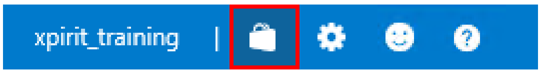
Exercise 6 “Creating the release”

Your machine is now prepared for IIS with all its features. Next we are going to setup a release pipeline.

1. Navigate to the VSTS environment
2. Navigate to the [RELEASE] hub
3. Create a new Release Definition using the wizard
4. Start with an empty template
5. Choose your [Build]
6. Check the [Continuous Deployment] box
7. Choose the [Default] agent queue.
8. The basic definition is now setup. Save it with a name [CentricRM**X**]

Now we need some addition task to be able to release our application. Navigate to the visual studio Marketplace from your VSTS environment.

1. This can be done by selecting the “bag” icon on the top right corner;

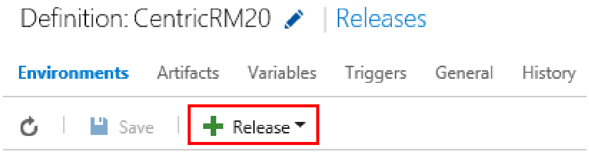


2. Search for the [IIS Web App Deployment Using WinRM] package
3. Choose [Install] from the package page.
4. Click [Continue] to add the package to your VSTS environment
5. Click [Confirm] to finish the installation.
6. Click [Close].
7. Navigate back to the [RELEASE] hub.
8. Edit the [CentricRM**X**] release definition
9. Navigate to [Environments]
10. Click [Add tasks]
11. From the [Deploy] section choose the [WinRM - IIS Web App Management] task.
12. Choose [Add] to add an instance and click [Close]
13. Use the following properties
    a. Machines = localhost
    b. Admin login = CentricAdmin
    c. Password = P@ssw0rd1
    d. Check “create or update website”
    e. Website name = CentricRMWeb**X**
    f. Check [Create or Update application pool]
    g. Application Pool name = CentricRMWeb**X**
14. From the [Deploy] section choose the [WinRM – IIS Web App Deployment] task.
15. Choose [Add] to add an instance and click [Close]

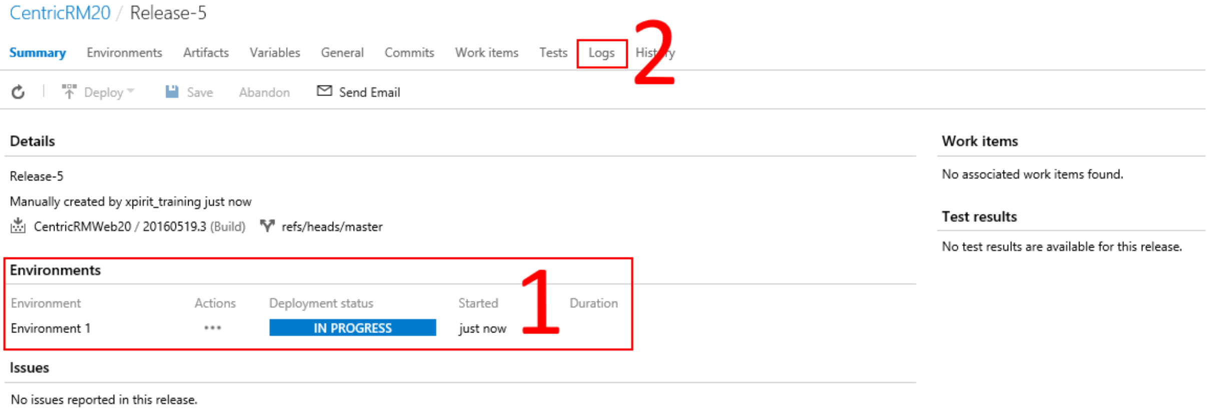
16. Use the following properties
    a. Machines = localhost
    b. Admin login = CentricAdmin
    c. Password = P@ssw0rd1
    d. Web deploy package =

       $(System.DefaultWorkingDirectory)\CentricRMWebX\drop\CentricRMWebX.zip

    e. Website Name = CentricRMWebX
17. Save the release definition
18. Create a new release



19. Select the latest build to release
20. Click [Create] to start the release
21. Navigate to the running release
22. Notice the status in section 1



Details

Release-5

Manually created by xpirit_training just now

CentricRMWeb20 / 20160519.3 (Build) refs/heads/master

Environments

Environment	Actions	Deployment status	Started	Duration
Environment 1	•••	IN PROGRESS	just now	

Issues

No issues reported in this release.

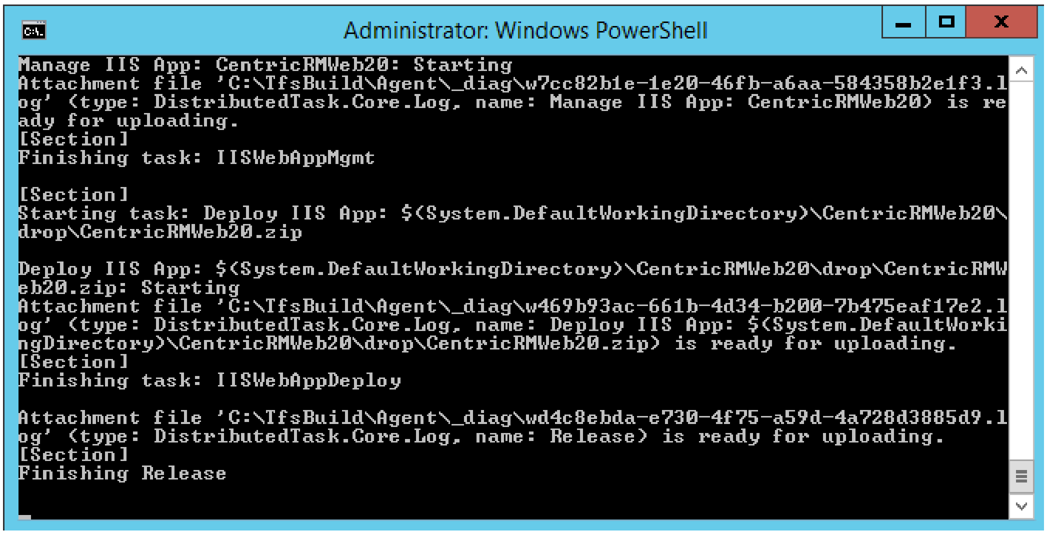
Work items

No associated work items found.

Test results

No test results are available for this release.

23. Navigate to [Logs] at location 2
24. In depth details can be monitored there.
25. Also check your Agents command window, this shows that it has been deploying your application.



Administrator: Windows PowerShell

```
Manage IIS App: CentricRMWeb20: Starting
Attachment file 'C:\TfsBuild\Agent\_diag\w7cc82b1e-1e20-46fb-a6aa-584358b2e1f3.l
og' (type: DistributedTask.Core.Log, name: Manage IIS App: CentricRMWeb20) is re
ady for uploading.
[Section]
Finishing task: IISWebAppMgmt

[Section]
Starting task: Deploy IIS App: $(System.DefaultWorkingDirectory)\CentricRMWeb20\
drop\CentricRMWeb20.zip

Deploy IIS App: $(System.DefaultWorkingDirectory)\CentricRMWeb20\drop\CentricRMW
eb20.zip: Starting
Attachment file 'C:\TfsBuild\Agent\_diag\w469b93ac-661b-4d34-b200-7b475eaf17e2.l
og' (type: DistributedTask.Core.Log, name: Deploy IIS App: $(System.DefaultWorki
ngDirectory)\CentricRMWeb20\drop\CentricRMWeb20.zip) is ready for uploading.
[Section]
Finishing task: IISWebAppDeploy

Attachment file 'C:\TfsBuild\Agent\_diag\wd4c8ebda-e730-4f75-a59d-4a728d3885d9.l
og' (type: DistributedTask.Core.Log, name: Release) is ready for uploading.
[Section]
Finishing Release
```

26. Browse to http://localhost:8080/ to see your application!