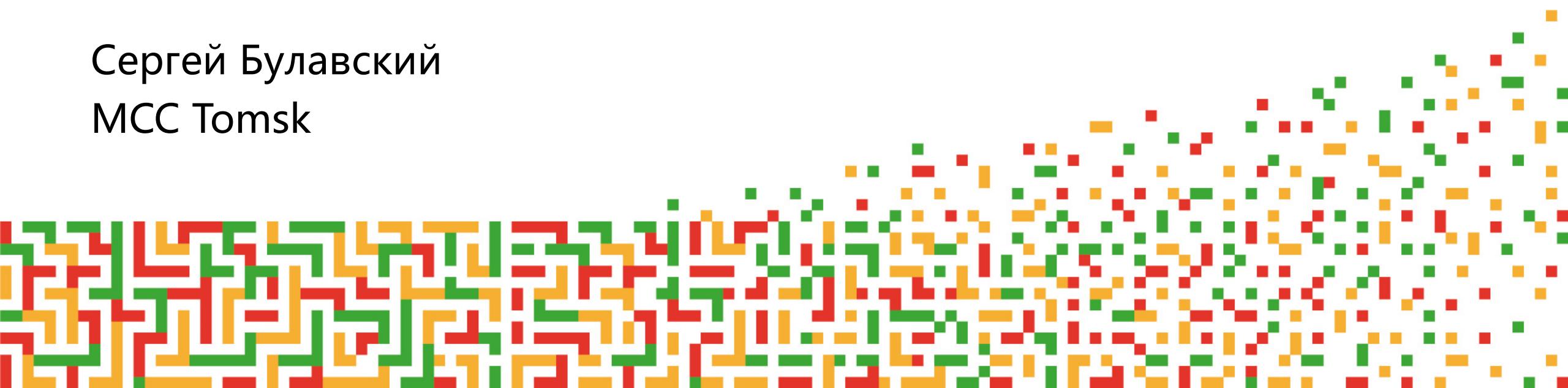


Build definition as a code with F#

Сергей Булавский
МСС Tomsk



Build definition in tfs

Agent job 1 +

Run on agent

- >  Use .NET Core sdk 2.2.101
.NET Core SDK Installer
- >  dotnet restore
.NET Core
- >  dotnet build
.NET Core
- >  dotnet test
.NET Core
- >  npm install
npm
- >  npm custom
npm
- >  dotnet publish
.NET Core
- >  Build an image
Docker
- >  Push an image
Docker

Build definition as a code with F#



Build definition teamcity

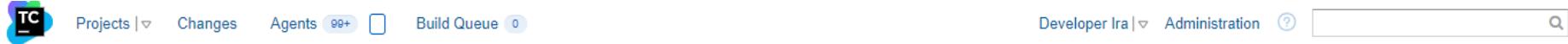
Based on [npm packages / npmPackageWithTest \(detach\)](#)

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. [?](#)

+ Add build step	Auto-detect build steps
Build Step	Parameters Description
1. run npm version <small>(inherited)</small>	Command Line Custom script: npm version %BuildNumber% --force Execute: If all previous steps finished successfully Edit ≡
2. run npm install <small>(inherited)</small>	Command Line Custom script: npm install Execute: If all previous steps finished successfully Edit ≡
3. run npm test <small>(inherited)</small>	Command Line Custom script: npm run teamcitytest Execute: If all previous steps finished successfully Edit ≡
4. run npm publish <small>(inherited)</small>	Command Line Custom script: #!/bin/bash (and 6 more lines) Execute: If all previous steps finished successfully Edit ≡

Teamcity build definition history



Projects | Changes Agents 90+ Build Queue 0 Developer Ira | Administration

Administration

Project-related Settings

- Projects
- All Builds
- Build Time
- Disk Usage
- Server Health
- Audit**

User Management

- Users
- Groups
- Roles

Audit

Show: in: Tools / Documentation Generator by: Filter Results per page: 50

[Permalink](#)

Date	User	Action	Comment
22 Jan 19 17:30	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 12:39	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 12:39	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 12:19	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 12:01	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 11:36	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated

Teamcity build definition history

edited "Tools / Documentation Generator" build configuration settings

Junior Sergey: VCS roots of 'Documentation Generator' build configuration were updated

Ignore whitespaces

```

Copy
44   </parameters>
45   </runner>
46   <runner id="RUNNER_45" name="make html for Test Protocol" type="simpleRunner">
47     <parameters>
48       <param name="script.content" value="npm run-script build-test" />
49       <param name="teamcity.build.workingDir" value="test-protocol-markup-generator" />
50       <param name="teamcity.step.mode" value="default" />
51       <param name="use.custom.script" value="true" />
52     </parameters>
53   </runner>
54   <runner id="RUNNER_56" name="Run wkhtmltopdf for Test Protocol" type="simpleRunner">
55     <parameters>
56       <param name="script.content" value="&quot;C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe&quot; --print-media-type --headless" />
57       <param name="teamcity.step.mode" value="default" />
58       <param name="use.custom.script" value="true" />
59     </parameters>
60   </runner>
61   <runner id="RUNNER_106" name="make html for VSS" type="simpleRunner">
62     <parameters>
63       <param name="script.content" value="npm run-script build-verification" />
64       <param name="teamcity.build.workingDir" value="test-protocol-markup-generator" />
65       <param name="teamcity.step.mode" value="default" />
66       <param name="use.custom.script" value="true" />
67     </parameters>
68   </runner>
69   <runner id="RUNNER_108" name="Run wkhtmltopdf for VSS" type="simpleRunner">
70     <parameters>
71       <param name="script.content" value="&quot;C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe&quot; --print-media-type --headless" />
72       <param name="teamcity.step.mode" value="default" />
73       <param name="use.custom.script" value="true" />
74     </parameters>
75   </runner>
76 </build-runners>
77 <vcs-settings>
78   <vcs-entry-ref root-id="Tools_TestProtocols" />
79 </vcs-settings>
80 <requirements>
81   <equals id="RQ_2" name="env.OS" value="Windows_NT" />
82 </requirements>
83 <build-triggers />
84 <cleanup />
85 </settings>
86 </build-type>
87

```

44 </parameters>
45 </runner>
46 <runner id="RUNNER_45" name="make html for Test Protocol" type="simpleRunner">
47 <parameters>
48 <param name="script.content" value="npm run-script build-test" />
49 <param name="teamcity.build.workingDir" value="test-protocol-markup-generator" />
50 <param name="teamcity.step.mode" value="default" />
51 <param name="use.custom.script" value="true" />
52 </parameters>
53 </runner>
54 <runner id="RUNNER_56" name="Run wkhtmltopdf for Test Protocol" type="simpleRunner">
55 <parameters>
56 <param name="script.content" value=""C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe" --print-media-type --headless" />
57 <param name="teamcity.step.mode" value="default" />
58 <param name="use.custom.script" value="true" />
59 </parameters>
60 </runner>
61 <runner id="RUNNER_106" name="make html for VSS" type="simpleRunner">
62 <parameters>
63 <param name="script.content" value="npm run-script build-verification" />
64 <param name="teamcity.build.workingDir" value="test-protocol-markup-generator" />
65 <param name="teamcity.step.mode" value="default" />
66 <param name="use.custom.script" value="true" />
67 </parameters>
68 </runner>
69 <runner id="RUNNER_108" name="Run wkhtmltopdf for VSS" type="simpleRunner">
70 <parameters>
71 <param name="script.content" value=""C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe" --print-media-type --headless" />
72 <param name="teamcity.step.mode" value="default" />
73 <param name="use.custom.script" value="true" />
74 </parameters>
75 </runner>
76 </build-runners>
77 <vcs-settings />
78 <requirements>
79 <equals id="RQ_2" name="env.OS" value="Windows_NT" />
80 </requirements>
81 <build-triggers />
82 <cleanup />
83 </settings>
84 </build-type>
85

Build definition in teamcity (kinda source)

```
13  </parameters>
14  <build-runners>
15    <runner id="RUNNER_118" name="run npm version" type="simpleRunner">
16      <parameters>
17        <param name="script.content" value="npm version %BuildNumber% --force" />
18        <param name="teamcity.step.mode" value="default" />
19        <param name="use.custom.script" value="true" />
20      </parameters>
21    </runner>
22    <runner id="RUNNER_116" name="run npm install" type="simpleRunner">
23      <parameters>
24        <param name="script.content" value="npm install" />
25        <param name="teamcity.step.mode" value="default" />
26        <param name="use.custom.script" value="true" />
27      </parameters>
28    </runner>
29    <runner id="RUNNER_117" name="run npm test" type="simpleRunner">
30      <parameters>
31        <param name="script.content" value="npm run teamcitytest" />
32        <param name="teamcity.step.mode" value="default" />
33        <param name="use.custom.script" value="true" />
34      </parameters>
35    </runner>
36    <runner id="RUNNER_119" name="run npm publish" type="simpleRunner">
37      <parameters>
38        <param name="script.content"><![CDATA[#!/bin/bash
39 branch=%teamcity.build.branch%
40 echo 'branch is '$branch
41 if [[ $branch == *"pull"* ]]; then
42   exit 0
43 fi
44 npm publish --registry %npm_FeedAddress%]]></param>
45        <param name="teamcity.step.mode" value="default" />
46        <param name="use.custom.script" value="true" />
47      </parameters>
48    </runner>
49  </build-runners>
```

Story time!

DevOps Vasya

8



- Powershell pro
- Many years of Windows
- Currently working with teamcity

Developer Ira

9



- Confident middle software developer
- C#/TypeScript pro
- Barely knows PS



- Very curious
- Doing before thinking
- What is powershell?
- Git push before leaving on friday evening

Working with history in teamcity

11

The screenshot shows the TeamCity Administration interface. On the left, the 'Audit' section is selected under 'Project-related Settings'. The main area displays a table of audit logs for 'Build configuration actions' related to 'Tools / Documentation Generator' by 'All users'. The logs show multiple edits made by 'Junior Sergey' on January 22, 2019, at various times between 11:37 and 17:30. Below this, a large portion of the TeamCity build definition XML code is shown, numbered from 60 to 87. A vertical scroll bar is visible on the right side of the code editor. A black arrow points from the bottom of the audit log table down towards the XML code. The XML code includes sections for runners, parameters, build runners, VCS settings, requirements, and cleanup triggers.

Date	User	Action	Comment
22 Jan 19 17:30	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated
22 Jan 19 12:39	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 12:39	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 12:19	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 12:01	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	parameters of 'Documentation Generator' build configuration were updated
22 Jan 19 11:37	Junior Sergey	Tools / Documentation Generator build configuration settings were edited (view change)	runners of 'Documentation Generator' build configuration were updated

```
60      </runner>
61    <runner id="RUNNER_106" name="make html for VSS" type="simpleRunner">
62      <parameters>
63        <param name="script.content" value="npm run-script build-verification" />
64        <param name="teamcity.build.workingDir" value="test-protocol-markup-generator" />
65        <param name="teamcity.step.mode" value="default" />
66        <param name="use.custom.script" value="true" />
67      </parameters>
68    </runner>
69    <runner id="RUNNER_108" name="Run wkhtmltopdf for VSS" type="simpleRunner">
70      <parameters>
71        <param name="script.content" value=""C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe" --print-media-type --headless" />
72        <param name="teamcity.step.mode" value="default" />
73        <param name="use.custom.script" value="true" />
74      </parameters>
75    </runner>
76  </build-runners>
77  <vcs-settings>
78    <vcs-entry-ref root-id="Tools_TestProtocols" />
79  </vcs-settings>
80  <requirements>
81    <equals id="RQ_2" name="env.OS" value="Windows_NT" />
82  </requirements>
83  <build-triggers />
84  <cleanup />
85  </settings>
86 </build-type>
87
```

```
60      </runner>
61    <runner id="RUNNER_106" name="make html for VSS" type="simpleRunner">
62      <parameters>
63        <param name="script.content" value="npm run-script build-verification" />
64        <param name="teamcity.build.workingDir" value="test-protocol-markup-generator" />
65        <param name="teamcity.step.mode" value="default" />
66        <param name="use.custom.script" value="true" />
67      </parameters>
68    </runner>
69    <runner id="RUNNER_108" name="Run wkhtmltopdf for VSS" type="simpleRunner">
70      <parameters>
71        <param name="script.content" value=""C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe" --print-media-type --headless" />
72        <param name="teamcity.step.mode" value="default" />
73        <param name="use.custom.script" value="true" />
74      </parameters>
75    </runner>
76  </build-runners>
77  <vcs-settings />
78  <requirements>
79    <equals id="RQ_2" name="env.OS" value="Windows_NT" />
80  </requirements>
81  <build-triggers />
82  <cleanup />
83  </settings>
84 </build-type>
85
```

Build definition as a code with F#



Why build scripts

12

- Versioning (branching, rolling back) and
 - Change tracking (Pull requests etc)
 - Local run/debug
 - Almost no build server configuration.
-
- Fake (F#)
 - Cake (C#)
 - PSAKE (Powershell)

- Open source
- Runs on Linux, Mac OS X, Android, IOS, GPU's (<http://fsharp/use/gpu>) and browsers (can generate JavaScript).
- .NET Compatible (Originated as Ocaml .NET Framework implementation)
- Functional-first
- Strongly typed

What is F#?

```
public class SomeClass
{
    public SomeClass(string firstField, int secondField)
    {
        FirstField = firstField;
        SecondField = secondField;
    }

    public string FirstField { get; private set; }
    public int SecondField { get; private set; }
}

public IEnumerable<TSource> Where<TSource>(
    IEnumerable<TSource> source,
    Func<TSource, bool> predicate
)
{
    // standard LINQ implementation
    return source.Where(predicate);
}

public IEnumerable<IGrouping< TKey, TSource>> GroupBy<TSource, TKey>(
    IEnumerable<TSource> source,
    Func<TSource, TKey> keySelector
)
{
    // standard LINQ implementation
    return source.GroupBy(keySelector);
}
```

C#

Build definition as a code with F#



What is F#?

```
public class SomeClass
{
    public SomeClass(string firstField, int secondField)
    {
        FirstField = firstField;
        SecondField = secondField;
    }

    public string FirstField { get; private set; }
    public int SecondField { get; private set; }
}

public IEnumerable<TSource> Where<TSource>(
    IEnumerable<TSource> source,
    Func<TSource, bool> predicate
)
{
    // standard LINQ implementation
    return source.Where(predicate);
}

public IEnumerable<IGrouping<TKey, TSource>> GroupBy<TSource, TKey>(
    IEnumerable<TSource> source,
    Func<TSource, TKey> keySelector
)
{
    // standard LINQ implementation
    return source.GroupBy(keySelector);
}
```

C#

Build definition as a code with F#

```
type SomeClass(firstField:string, secondField:int) =
    member this.FirstField = firstField
    member this.SecondField = secondField

let Where source predicate =
    Seq.filter predicate source

let GroupBy source keySelector =
    Seq.groupBy keySelector source
```

F#

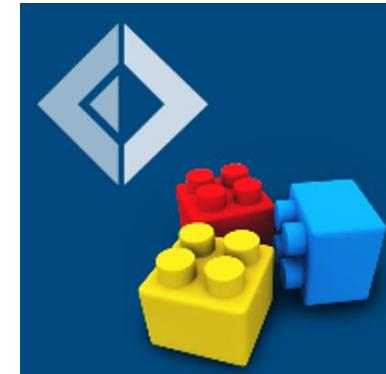
```
1 type SomeClass(firstField:string, secondField:int) =
2     member this.FirstField = firstField // string
3     member this.SecondField = secondField // int
4
5 let Where source predicate = // seq<'a> -> ('a -> bool) -> seq<'a>
6     Seq.filter predicate source
7
8 let GroupBy source keySelector = // seq<'a> -> ('a -> 'b) -> seq<'b * seq<'a>
9     Seq.groupBy keySelector source
```



What is FAKE?

16

Fake (F# Make) - Open source DSL (based on F#) for building, copying, running tests, executing applications, and more.



Versions:

FAKE 4

FAKE 5

Why FAKE?

- Open source
- Community (Mainly C# developers)
- .NET/.NET Core
- Easy to use even without F# experience
- If Tools is not enough for you - write your own.

The screenshot shows the GitHub repository page for fsharp/FAKE. At the top, two large boxes highlight '8,777 commits' and '308 contributors'. Below this, the repository name 'fsharp / FAKE' is shown along with a star count of 957 and a fork count of 562. A red box highlights the '8,777 commits' button. To the right, the text 'v. 5.12.1' is displayed in red. The main repository page includes tabs for Code, Issues (33), Pull requests (10), Projects (1), and Insights. Below the tabs, it says 'FAKE - F# Make https://fake.build'. The commit count and contributor count are also highlighted in red on the main stats bar. A red box highlights the '308 contributors' button. The bottom of the page shows a release note from 'matthid' and a green 'Clone or download' button.

Since 08/10

The screenshot shows the GitHub repository page for cake-build/cake. At the top, two large boxes highlight '2,582 commits' and '161 contributors'. Below this, the repository name 'cake-build / cake' is shown along with a star count of 2,092 and a fork count of 518. A red box highlights the '2,582 commits' button. To the right, the text 'v. 0.32.1' is displayed in red. The main repository page includes tabs for Code, Issues (293), Pull requests (26), Projects (1), Wiki, and Insights. Below the tabs, it says 'Cake (C# Make) is a cross platform build automation system. https://cakebuild.net/'. The commit count and contributor count are also highlighted in red on the main stats bar. A red box highlights the '161 contributors' button. The bottom of the page shows a merge pull request from 'mholo65' and a green 'Clone or download' button.

Since 05/14

Syntax of .fsx

```
#load ".fake/build.fsx/intellisense.fsx" // Loads intellisense for current dependencies.

open Fake.Core                                // Opens specific namespace. (usings)
open Fake.DotNet
open Fake.IO
open Fake.IO.Globbing.Operators

Target.create "Clean" (fun _ ->
    !! "src/**/bin"                          // Find all matching pattern.
    ++ "src/**/obj"                         // Include all matching additional pattern.
    -- "src/**/do-not-delete"                // Exclude everything found by this pattern.
    |> Shell.cleanDirs                      // Clean all directories that will be found.
)

Target.create "Build" (fun _ ->
    !! "src/**/*.*proj"
    |> Seq.iter (DotNet.build id)
)

Target.create "All" ignore

"Clean"
  ==> "Build"                                // Can also be "Clean" ==> "Build" ==> "All"
  ==> "All"

Target.runOrDefault "All"                      // What to run by default when script starts.
```

```
"Build"
    ==> "NUnitTest"
    => ("xUnitTest", hasBuildParam "xUnitTest")
    ==> "Deploy"
```

Build dependencies and their relations

```
"CleanDotNetProjects"
  ==> "BuildDotNetProjects"
  ==> "UnitTests"
  ==> "NpmCleanInstall"
  ==> "NpmBuild"
  ==> "RunSeleniumTests"
  ==> "RunIntegrationTests"
  ==> "CheckNpmPackages"
  ==> "CI"
  ==> "PublishArtifacts"
```

Shortened DependencyGraph for Target PublishArtifacts:

```
<== PublishArtifacts
  <== CI
    <== CheckNpmPackages
      <== RunIntegrationTests
        <== RunSeleniumTests
          <== NpmBuild
            <== NpmCleanInstall
              <== UnitTests
                <== BuildDotNetProjects
                  <== CleanDotNetProjects
```

The running order is:

```
Group - 1
  - CleanDotNetProjects
Group - 2
  - BuildDotNetProjects
Group - 3
  - UnitTests
Group - 4
  - NpmCleanInstall
Group - 5
  - NpmBuild
Group - 6
  - RunSeleniumTests
Group - 7
  - RunIntegrationTests
Group - 8
  - CheckNpmPackages
Group - 9
  - CI
Group - 10
  - PublishArtifacts
```

Build dependencies and their relations (parallel)

21

```
"CleanDotNetProjects"
  ==> "BuildDotNetProjects"
  ==> "UnitTests"

"NpmCleanInstall"
  ==> "NpmBuild"

"UnitTests"
  ==> "RunIntegrationTests"
  ==> "CI"

"NpmBuild"
  ==> "RunSeleniumTests"
  ==> "CI"

"CheckNpmPackages" ==> "CI"
"CI" ==> "PublishArtifacts"
```

Shortened DependencyGraph for Target PublishArtifacts:

```
<== PublishArtifacts
  <== CI
    <== RunSeleniumTests
      <== NpmBuild
        <== NpmCleanInstall
          <== RunIntegrationTests
            <== UnitTests
              <== BuildDotNetProjects
                <== CleanDotNetProjects
                  <== CheckNpmPackages
```

The running order is:

- Group - 1
 - NpmCleanInstall
 - CleanDotNetProjects
 - CheckNpmPackages
- Group - 2
 - NpmBuild
 - BuildDotNetProjects
- Group - 3
 - RunSeleniumTests
 - UnitTests
- Group - 4
 - RunIntegrationTests
- Group - 5
 - CI
- Group - 6
 - PublishArtifacts

Super complicated build server config? nope

22

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. [?](#)

[+ Add build step](#) [Reset step order](#) [Auto-detect build steps](#)

Build Step	Parameters Description	
1. Run FAKE (inherited, overridden)	Command Line Custom script: IF NOT EXIST ".fake\fake.exe" ((and 3 more lines) Execute: If all previous steps finished successfully	Edit ≡

Build Step

Runner type: Simple command execution

Step name: Optional, specify to distinguish this build step from other steps.

Run:

Custom script: *
Enter build script content:
`IF NOT EXIST \".fake\fake.exe\" (\n dotnet tool install fake-cli --tool-path ./fake\n)\n.fake\fake.exe run build.fsx`

A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.

Build definition as a code with F#



Ionide-vscode-fake extension + Visual Studio Code

Visual Studio for windows



So what about next Friday evening?

24

Sergey could've:

- Experiment with build locally, in a branch.
- Run everything locally.
- Leave progress in his own git branch.

Ira could've:

- Easily roll back to last working version in git.
- Track changes that was made by sergey.
- Debug everything locally without taking any build agents.

Спасибо. Вопросы?

F# Make - <https://fake.build/>

F# Foundation - <https://fsharp.org/>

F# Foundation Slack - <https://fsharp.slack.com>



До встречи 21 марта на
TomskDotNet #3!

Точка Кипения
21 марта, 18:30