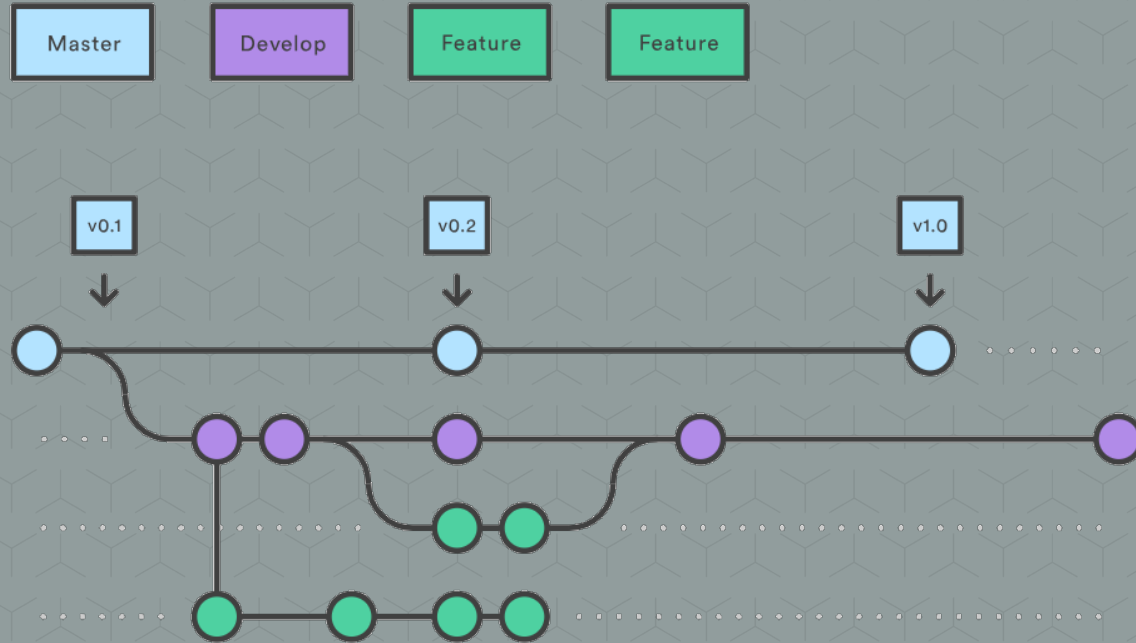

Feature branches in TC and how to use them in our CI stack

Feature branches



Prerequisite – fitting into our streams setup

TestingPWS [2 info items](#)

General Settings

VCS Roots 2

Report Tabs

Parameters 16

Connections

Shared Resources ✓

Meta-Runners

Maven Settings

Issue Trackers

Cloud Profiles

Clean-up Rules

Versioned Settings ✓

Artifacts Storage

NuGet Feed

WebHooks

SSH Keys

Suggestions 4

◀ Hide unconfigured

Last edited 5 hours ago
by Daniele Iasella (view history)

Settings are stored in VCS (view history)

Versioned Settings

Configuration

Change Log

On this page you can enable synchronization of the current project settings with the version control: if the project settings are changed, the affected configuration files will be checked in to the version control; if the configuration files are changed in the version control, the changes will be applied to the project. Note that the passwords which are configured in the project and subprojects (e.g. in project's VCS roots) are stored in the configuration files and can be exposed this way. [?](#)

Supported version control systems: **Team Foundation Server**, **Git**, **Mercurial**, **Subversion**, **Perforce**.

☐ Use settings from a parent project

☐ Synchronization disabled

☒ Synchronization enabled

Project settings VCS root:

PWS_featurebranches

Edit VCS root

When build starts: [?](#)

☐ always use current settings

Builds use current project settings from the TeamCity server. Settings changes in branches, history and personal builds are ignored.

☐ use current settings by default

Builds use current project settings from the TeamCity server. Users can run a build with settings from VCS via the run custom build dialog.

☒ use settings from VCS

Builds in branches and history builds use settings from VCS. Note that users can change settings in **personal** builds from IDE.

☒ Show settings changes in builds

☒ Store secure values (like passwords or API tokens) outside of VCS

Settings format:

xml

Current Status:

[15:46:39]: Changes from VCS are applied to project settings, last change 'TeamCity change in 'Digital Engineering / TestingPWS' project: triggers of 'Create Octopus Release' build configuration were updated', revision 15a03c0995fa6d85e5cbb3bdabd9dc1f17c84976 [?](#)

abcam

Daniele Iasella

3

Feature branches functionality

Administration / <Root project> / Digital Engineering / TestingPWS Action

Edit VCS Root

Type of VCS

Type of VCS:

VCS Root

VCS root name:
A unique name to distinguish this VCS root from other roots.

VCS root ID: Regenerate ID
VCS root ID must be unique across all VCS roots. VCS root ID can be used in parameter references to VCS root parameters and REST API.

General Settings

Fetch URL:
It is used for fetching data from the repository.

Push URL:
It is used for pushing tags to the remote repository. If blank, the fetch url is used.

Default branch:
The main branch or tag to be monitored

Branch specification:
Edit branch specification:

Branches to monitor besides the default one as a newline-delimited set of rules in the form of `+|-branch name` (with the optional `*` placeholder)

Use tags as branches: ☐ Enable to use tags in the branch specification

[View usages](#)
[Last edited 2 days ago by Daniele Iasella \(view history\)](#)
[Settings are stored in VCS \(view history\)](#)

Teamcity UI ----- finally!

Digital Engineering Edit Project Settings

TestingPWS <Active branches>

Testing11

Overview Change Log Statistics Current Problems Investigations Muted Problems Build Chains Flaky Tests WebHooks

☐ Hide successful configurations

Build

test	#0.0.3.109	Tests passed: 3278	Daniele Iasella (2)	4 hours ago (7m:39s)
f1	#0.0.1.104	Tests passed: 3278	Changes (10)	one day ago (22m:01s)
f2	#0.0.2.87	Tests passed: 3176	Daniele Iasella (1)	3 days ago (5m:52s)

Build Functional Tests

test	#0.0.3.109	Success	Daniele Iasella (2)	4 hours ago (1m:35s)
f1	#0.0.1.104	Success	Changes (10)	one day ago (2m)
f2	#0.0.2.87	Success	Daniele Iasella (1)	3 days ago (1m:06s)

Create Octopus Release

Problem with Finish Build Trigger Details

test	#0.0.3.106	Success	Pending (57)	22 hours ago (1m:39s)
f1	#0.0.1.104	Success	Changes (51)	one day ago (5m:12s)
f2	#0.0.2.87	Success	Changes (10)	one day ago (1m:13s)
<default>	#0.0.0.86	Success	Daniele Iasella (1)	3 days ago (1m:25s)

Deploy CI

f1	#0.0.1.104	Success	Pending (100+)	one day ago (14m:01s)
f2	#0.0.2.87	Success	Changes (10)	3 days ago (13m:46s)
<default>	#0.0.1.52	Success	Daniele Iasella (1)	7 days ago (10m:33s)

Integration Test CI

Sean Lindsay-Smith is assigned to investigate the build configuration

			Pending (100+)	
--	--	--	----------------	--

Avoid triggers from F1/F2/P1 to Environments not in the channel

The screenshot shows the Jenkins 'Promote Performance' configuration page. The left sidebar contains a navigation menu with options like 'General Settings', 'Version Control Settings', 'Build Step: OctopusDeploy: Promot...', 'Triggers 1', 'Failure Conditions', 'Build Features 1', 'Dependencies 1', 'Parameters 18', 'Agent Requirements', 'Suggestions', and '< Hide unconfigured'. The main content area is titled 'Triggers' and includes a '+ Add new trigger' button and a 'Finish Build Trigger' section. A modal dialog box titled 'Finish Build Trigger' is open, showing the configuration for a new trigger. The dialog includes a 'Build configuration' dropdown set to 'Digital Engineering / PWS / Promote REG', a warning message 'There is no snapshot dependency on the selected build configuration.', and a checked checkbox for 'Trigger after successful build only'. The 'Branch Filter' section is also visible, with a text area containing the DSL rules: `+:<default>`, `-:tc/f1`, and `-:tc/f2`. The dialog has 'Save', 'Cancel', and 'View DSL' buttons at the bottom.

Administration / <Root project> / Digital Engineering / TestingPWS

Run ... Actions Build Configuration Home

Promote Performance

General Settings
Version Control Settings
Build Step: OctopusDeploy: Promot...
Triggers 1
Failure Conditions
Build Features 1
Dependencies 1
Parameters 18
Agent Requirements
Suggestions
< Hide unconfigured

Last edited 2 hours ago
by Daniele Iasella (view history)

Settings are stored in VCS (view history)

View DSL

Triggers

Triggers are used to add builds to the queue either when an event occurs (like a VCS check-in) or periodically with some configurable interval.

+ Add new trigger

Trigger

Finish Build Trigger

Build configuration: Digital Engineering / PWS / Promote REG

⚠ There is no snapshot dependency on the selected build configuration.

☒ Trigger after successful build only

Branch Filter

Branch filter:

Edit Branch Filter:

```
+:<default>
-:tc/f1
-:tc/f2
```

Newline-delimited set of rules in the form of +|-:logical branch name (with an optional * placeholder)

Save Cancel View DSL

Pros

- Unique project regardless number of *streams / branches*
- Better visibility of all the streams
- Easier start from a template (set once and reuse without resetting all streams)
- Auto-merge and merge request
- Revert changes in TC projects!!! (Not related to the feature branches functionality but to Versioned Settings instead)

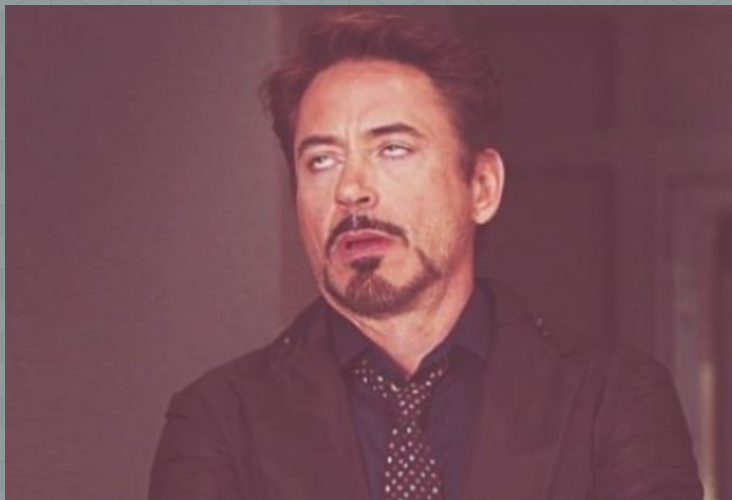
Cons?

TC's project UI driven by <default> branch (other branches?)



Cons?

TeamCity's `%build.counter%` variable is **unique** per project



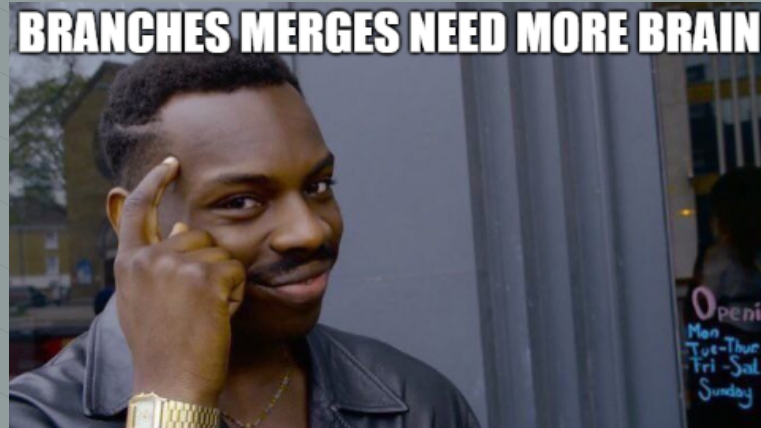
Cons?

Manual run could be red if they run *wrong step* per branch



Cons?

Merging branches in master will cause conflicts - [.teamcity/](https://www.teamcity.com/)*



Future spike

- Setup “Commit Status Publisher” build feature in TC for Gitlab merge request
- Auto-merge of feature branches in *stream/channel* branches

So....
Shall we try?