



Fork me on GitHub

PRACTICAL

Git and GitHub

Based on blog posts by

Dinis Cruz

beta version

Practical Git and GitHub

Real-world examples of solutions for problems that occurred during day to day Git/GitHub usage.

Dinis Cruz

Dieses Buch wird verkauft, unter http://leanpub.com/Practical_Git

Diese Version wurde veröffentlicht am 2016-02-06



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.



This work is licensed under a [Creative Commons Attribution 3.0 Unported License](#)

Twitter dieses Buch!

Bitte unterstütz Dinis Cruz, indem du über dieses Buch auf [Twitter](#) weiterempfiehlst!

Vorschlag: Verwende den folgenden Hashtag, wenn du über dieses Buch twitterst:
[#PracticalGit](#).

Was sagen Andere über dieses Buch? Klick hier, um nach diesem Hashtag auf Twitter zu suchen:

<https://twitter.com/search?q=#PracticalGit>

Contents

Leanpub book, originally based on Blog posts	i
Change log	iii
1. October 2012	1
Using a Git Branch to fix a Bug	2
Git and GitHub commands to create and deploy new version of TeamMentor	4
Idea: Sync Blogger Posts with a GitHub repository	13
Approving a GitHub Pull Request Workflow	14
Deploying TeamMentor to AppHarbor (.NET Cloud) using Git push	20
The need to create forks/clones for TeamMentor.net website	26
Using Git Branches to deal with the multi-config variations of TeamMentor	29
Handling content changes made on hosted site created by Git clone (with auto Git commits and pushes)	31
Going back in time using Git's checkout	33
Adding Tags to TeamMentor Master repository	37
Creating the final TeamMentor with SI Library repository via multiple Git pulls and pushes	41
2. November 2012	65
Pretty cool visualization of the ‘GitHub based’ TeamMentor Development+QA+Release workflow	66
3. December 2012	68
Comparing two GitHub Issues List	69
Using TeamCity to build on Git Commit, deploy to AppHarbor and open browser	71
Minimum required files to run git.exe on windows (for clone, push and pull)	76
Rewriting Git History (locally and at GitHub)	77
4. January 2013	80
GitHub is having some probs today	81

CONTENTS

Dangerous bug between Git, GitHub and Windows (duplicate directories with different capitalization)	84
Can Git be used instead of Word's 'Track Changes'	89
5. March 2013	90
Seeing an NGit Diff by using reflection to access the internal Sharpen.ByteArrayOutputStream Class	91
GitHub.com needs to improve their the 'Normal' status definition and error reporting dashboard	98
Another GitHub 'Normal' status that doesn't allow me to Push	100
Prob with (older version of) NGit where it was failing to create Git repositories in Azure/TeamCity	102
Creating a version TeamMentor which uses the new GitUserData.config file	107
Changing the 'View TM article by anonymous users' status via GitHub	123
Creating a new TeamMentor test site using TeamCity, GitHub and Azure	131
Using Git Branches to fix Issues added to TeamMentor's GitHub repository	146
Extracting content files from a Azure deployed version of TeamMentor (pre 3.3 git support)	151
Using NGit to create native Git support in Azure deployed app (with automatic pushes and pulls)	170
6. April 2013	179
Git pulling a TeamMentor Library and renaming it	180
Creating QA versions of TeamMentor UserData repository, and using branches to show/test the multiple config options	192
Changing a User's ExpiryDate from GitHub hosted file	205
Linus gift to the world will be Git not Linux (and what about an OS built on top of an hash-driven file system?)	209
What the move from HTML to WikiText looks like (in GitHub)	210
Is Git a Single point of failure for TeamMentor?	214
Setting up Ian's CI Development Environment (for TeamMentor)	216
7. May 2013	227
Example of two TeamMentor sites using the same GitHub Content Library	228
AzureGate - how Azure's 'subscription upgrade' crazy mode caused us to stop using Azure for VM Hosting (and Git+GitHub saved the day)	236
Great presentation on Git Branching (very similar to the model we are using in TeamMentor)	239

CONTENTS

Fixing bug in TBot user editor via Git merge of fix developed on another repo's branch	240
Releasing HotFix 1 for TeamMentor 3.3 (using Git to deploy updates to live servers)	248
Great post - Git: Who cares about branches? It's all about collaboration and code reviews	259
Creating o2platform.com website using GitHub Pages (with screenshots of all design options)	260
8. June 2013	275
Creating TeamMentor release 3.3.2 (3.2 version with HotFix 2)	276
Fixing a couple bugs and pushing new TeamMentor 3.4 Dev Version (from 4 to 5)	288
Gource Visualization of "TeamMentor Git Development - 18 Months in 180 Seconds"	292
9. August 2013	293
Creating a clone of WebGoat on GitHub	294
10. September 2013	302
Git Flow - Moving patches from one Commit into another Commit	303
Example of using GitHub Pull Requests to merge changes made on Branches	340
Script to Git Clone 13 repositories in order to have all TeamMentor Libraries in one folder	365
11. October 2013	371
Fixing the Merge conflict caused by one extra commit on TeamMentor master	372
Enabling GitHub Two Factor Authentication	390
Syncing all releases to the same commit and Tag (for TeamMentor v3.4)	395
12. January 2014	407
How to update a forked GitHub repo (in this case tm-sme/Lib_Vulnerabilities)	408
Updating the GitHub repos for the 1.6.0 release of the Eclipse Fortify Plugin	411
Updating GitHub Forks with latest commits from GitHub's 'parent' repo	417
Using TeamMentor 3.4 TBot admin pages to load and sync a Library hosted on GitHub	426
Using TeamMentor 3.4 TBot admin pages to load and sync UserData with a GitHub hosted repo	441
Adding files to TeamMentor's web root via a UserData folder (synced with GitHub)	457
13. February 2014	465
Reverting changes mades to TeamMentor articles	466

Leanpub book, originally based on Blog posts

This book started with the Git/Github related blog posts published on [my blog¹](#) and eventually evolved into the book you have in your hands (or eReader).

The idea to use my blog posts was to:

- kickstart the creation of this book (and provide a number of chapters that could be further improved)
- capture the multiple experiments and learning curves that I experienced (while learning how to use Git/Github)
- provide a ‘real world’ point of view on how to learn and use Git/Github

The theme of this book is ‘Practical’ (hence the name ‘Practical Git and GitHub’), and what I am trying to do here is to show how to use these technologies in practical situations, specially how to apply them to solve real problems. This means that sometimes the end result is a bit messy, with lots of rabbit-holes/tangents followed (and un-followed). But that is how it works. When learning a new technology, what is really important is to experiment a lot, and see how it works for the problem we have at hand.

Notes about current structure

At the moment, the chapter order is the one created by the original ‘import from blogger’ (i.e. by publish order). A better idea might be to create logical groups, so that the posts are ordered based on some user-friendly criteria (still to be defined)

At the moment I’m working/thinking about the best way to structure this content, and how to present it in a easy to read/consume format.

It would be great if you (the reader) could provide some feedback on the book’s structure, for example:

- if you think the book’s order should be different (or chapters renamed)
- there is content missing (really important to cover in an Git/Github book)
- a particular chapter is not very clear, easy-to-understand or relevant
- etc...

¹<http://blog.diniscruz.com>

About the Author

Dinis Cruz is a Developer and Application Security Engineer focused on how to develop secure applications. A key drive is on ‘Automating Application Security Knowledge and Workflows’ which is the main concept behind the OWASP O2 Platform and the FluentSharp APIs.

After many years (and multiple roles) Dinis is still very active at OWASP, currently leading the OWASP O2 Platform project and helping out other projects and initiatives.

After failing to scale his own security knowledge, learned Git, created security vulnerabilities in code published to production servers, delivered training to developers, and building multiple CI (Continuous Integration) environments; Dinis had the epiphany that the key to application security is “Secure Continuous Delivery: Developer’s Immediate Connection to What They’re Creating”. This ‘Immediate Connection/Feedback’ concept is deep rooted in the development of the O2 Platform, and is something that will keep Dinis busy for many years.

Change log

Here are the changes made (per version):

- **v0.41 (Jan 2016)**
 - Fixed bug where only October-2012 files were published
- **v0.40 (Jan 2016)**
 - Added new local editor ui (from o2platform/leanpub-book-site repo) which provides an local preview of the book as a website
 - renamed all files and saved them with an .md extension (most were still in .txt format)
 - a local site (using express) to preview content
 - generation of Leanpub manuscript folder from a better formatted folder structure
 - Fixed tons of content formatting issues and broken images
 - Fixed issues:
 - * Convert manuscript content to better folder structure #8²
 - * The images on article are completely wrong #7³
 - * Why .txt instead of .md in /manuscript/? #3⁴
- **v0.12 (April 2014)**
 - renamed number of files to have the name of the post (with spaces) and with MD extension
 - added Table_of_Contents.md file
- **v0.10 (March 2014)**
 - First release of book with raw import from blogger posts (no formatting or editing done)
 - Created Git repo on local dropbox sync folder
 - created github repository for this book: https://github.com/DinisCruz/Book_Practical_Git
 - Added this change log

²https://github.com/DinisCruz/Book_Practical_Git/issues/8

³https://github.com/DinisCruz/Book_Practical_Git/issues/7

⁴https://github.com/DinisCruz/Book_Practical_Git/issues/3

1. October 2012

- Using a Git Branch to fix a Bug
- Git and GitHub commands to create and deploy new version of TeamMentor
- Idea: Sync Blogger Posts with a GitHub repository
- Approving a GitHub Pull Request Workflow
- Deploying TeamMentor to AppHarbor (.NET Cloud) using Git push
- The need to create forks/clones for TeamMentor.net website
- Using Git Branches to deal with the multi-config variations of TeamMentor
- Handling content changes made on hosted site created by Git clone (with auto Git commits and pushes)
- Going back in time using Git's checkout
- Adding Tags to TeamMentor Master repository
- Creating the final TeamMentor with SI Library repository via multiple Git pulls and pushes

Using a Git Branch to fix a Bug

On Git I don't tend to use branches for big code changes, and the only time I tend to use them (on development) is when trying new code changes or fixes.

For example I was working last night on a CSRF fix for TeamMentor (more on that later) and I created a new branch using (`$ git checkout -b crst_test`) which contained a number of temp commits, like the ones that selectively disabled some Admin security demands so that I could

[Master / Commit History](#)

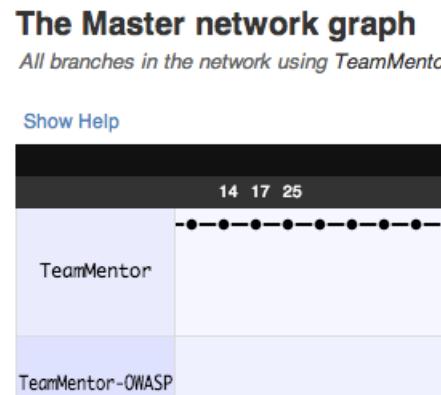
Oct 03, 2012

- Adding fix for CSRF bug**
DinisCruz authored 10 hours ago
- removing gzip compression to help debugging via direct requests**
DinisCruz authored 12 hours ago
- removing another admin demand**
DinisCruz authored 12 hours ago
- tmp removing some Admin demands**
DinisCruz authored 12 hours ago
- Re-adding the CSRF protection that was temporarily removed during App...** [...]
DinisCruz authored 12 hours ago

debug the issue better :): {width=512px}

As you can see by the Commits, I did a number of changes that were only pushed to the `crst_test` branch (`$ git push origin crst_test:crst_test`) and not be propagated to the main code base.

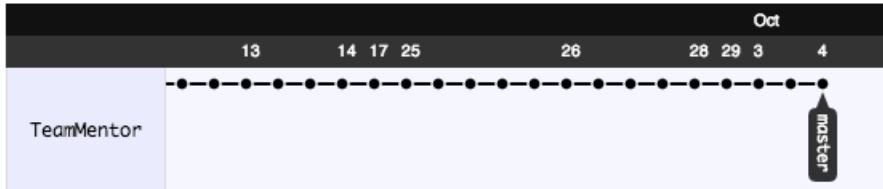
Then once I finally found the issue and fixed (see last commit), I went back to the the main branch (`$ git checkout master`) added my fixes, commit them and pushed them to the master branch (`$ git push origin master`).



At the moment this is what the main repo looks like: {width=612px}

The final step is to remove the local branch (`$ git branch -D crst_test`) and the branch from GitHub (`$ $ git push tm_master --delete crst_test`), which after GitHub recreates the graph (sometimes it takes a couple minutes¹), it looks like this: {width=712px}

[Show Help](#)



¹<http://stackoverflow.com/questions/12729317/force-github-network-graph-refresh>

Git and GitHub commands to create and deploy new version of TeamMentor

With the new Release of TeamMentor, I needed to upgrade the main GitHub repositories to the new version. This is the detailed version of the Git and GitHub workflow used.

My objective was to create a set-up where I could push the new code base (3.2) while keeping a clean separation with the previous one(s).

My solution was to use branches to hold each version (yes I could have used tags, but I prefer the isolation provided by branches).

The actions taken are:

- Identify the repository with the latest version of TeamMentor (the Code)
- Identify the repository with the TeamMentor Library to use (the Content)
- Identify the repository to upgrade (with both the Code and Content)
- Download the repository to upgrade and store it on a separate branch (the Backup)
- Download the Code repository
- Download and merge the Content repository
- Publish changes

More specifically (using some Git commands)

- Set up tm_master remote with the latest version of source-code (for example):
<https://github.com/TeamMentor/Master>²
- Set up tm_library remote with the TeamMentor library to use (for example):
https://github.com/TeamMentor-OWASP/Library_OWASP³
- Set up origin remote with the version to upgrade, which is the Code+Content version (for example):
<https://github.com/TeamMentor-OWASP/Master>⁴
- pull origin into master
- move master branch into master_Old.Version (for example) master_3_1
- create a new ‘orphan’ branch called master (and delete all files from it)
- pull tm_master into master
- pull (using tree merge strategy) tm_library into the Library_Data/XmlDatabase/TM_Libraries folder
- push all branches to origin

²<https://github.com/TeamMentor/Master>

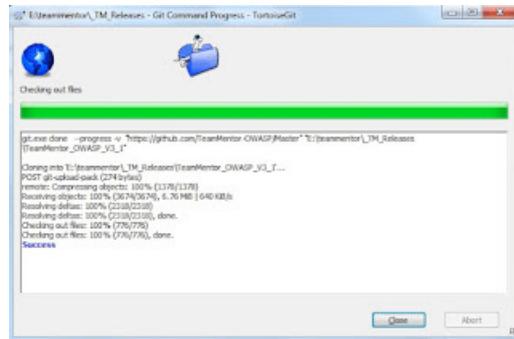
³https://github.com/TeamMentor-OWASP/Library_OWASP

⁴<https://github.com/TeamMentor-OWASP/Master>

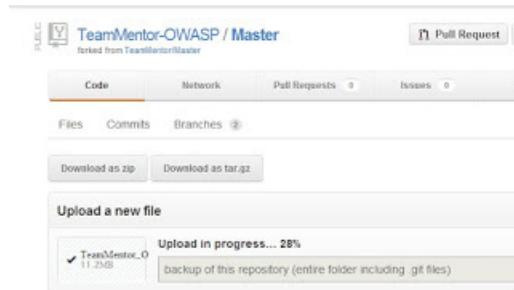
Step-by-Step workflow:

1) Create a local clone from the version to upgrade and back it up:

In this case from <https://github.com/TeamMentor-OWASP/Master>⁵

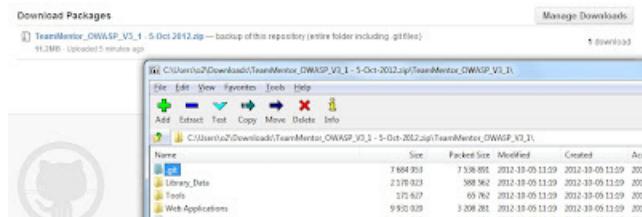


Once the repository is created locally, zip it up and upload it to the GitHub repository download section. It is important that this is the version with the .git folder (and not the .zip you can download from the repository main page). The fact that we can easily create a full backup of a complete Git repository is one of my favorite features of Git, and it allows for an easy restore for the inevitable mistakes that will be made :)



Tip: make sure to test the download, since I've seen cases where the upload fails and even if there is a download link on the page, there is no file downloadable:

⁵<https://github.com/TeamMentor-OWASP/Master>



Once you're happy that you have a backup, you can delete the folder locally created (the one you zipped)

2) Create a local repository with new version:

Start by creating a folder to hold the files (in this case *E:\teammentor_TM_Releases\TeamMentor_with_OWASP_Library*)



Open a git bash and cd into that folder

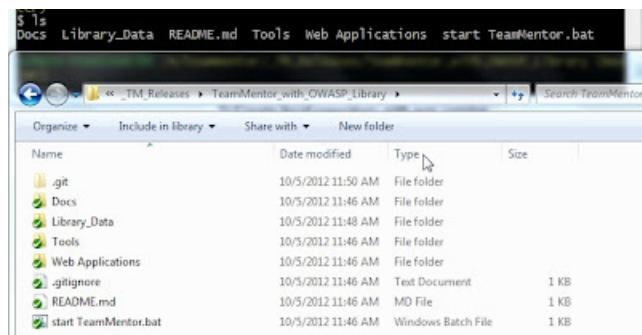
```
MINGW32/e/teammentor/_TM_Releases/TeamMentor_with_OWASP_Library
Welcome to Git (version 1.7.10-preview20120409)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

z02@WIN-FGMQ5AARJ80 ~
$ cd "E:\teammentor\_TM_Releases\TeamMentor_with_OWASP_Library"
z02@WIN-FGMQ5AARJ80 /e/teammentor/_TM_Releases/TeamMentor_with_OWASP_Library
$
```

Execute the following commands in sequence:

After completion the target folder should look like this:



To quickly test TeamMentor, double click on '*start TeamMentor.bat*'



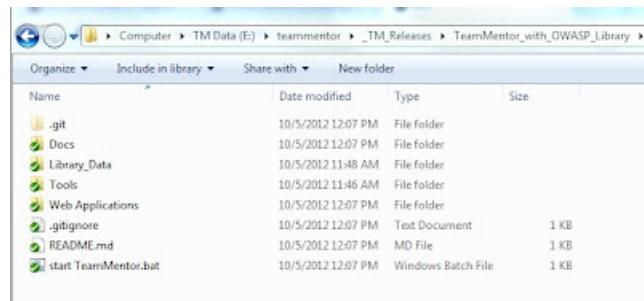
3) Test rolling back to an earlier version:

What is really cool with this set-up (and the power of Git), is that it is very easy to change to an earlier version. Leaving the webserver on (which is running from the *ToolsCassiniDev_4.0.ex* folder), go to the git batch and run the command: `$ git checkout -f TeamMentor_3_1`

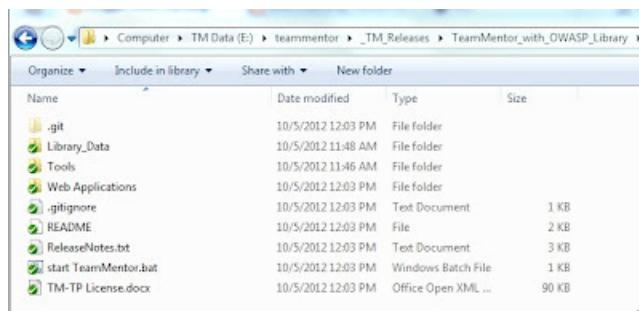
```
$ git branch
* master
* TeamMentor_3_1
$ git checkout -f TeamMentor_3_1
Switched to branch 'TeamMentor_3_1'
$
```

What this will do is to replace the current files in the target dir with the files from the TeamMentor_3_1 branch (which are the files from the repository we are upgrading).

This is what the file system looked like before the : `$ git checkout -f TeamMentor_3_1` command



And this is after



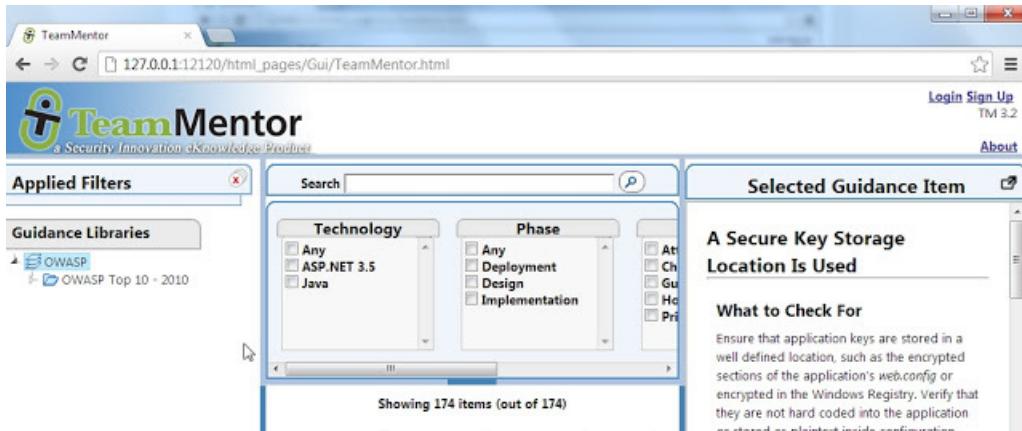
Can you spot the differences?

I can't overstate how powerful this is, specially due to the speed that it happens. What we are looking at here is a virtual file-system based on a Git database. It was this feature that really made me have a massive paradigm shift, and realize that **Git gives us a Version Control File-System** (maybe one day we will have an OS-Level git based file-system).

And just to make sure that we did change into the previous version of TeamMentor, refresh the browser and reopen <http://127.0.0.1:12120>⁶ (which will be the 3.1 version of TM)

Now go back to the Git Bash, switch to the master branch (via `$ git checkout -f master`) and refresh the browser:

⁶<http://127.0.0.1:12120/>



And we're back to version 3.2 (note the version number on the top right, under the 'Sign Up' link).

I have to say that I never get tired of seeing this '*virtual file system created by branches*' in action. Try the checkouts a couple times with a windows explorer window open (so that you see the files change in real time).

4) Confirm that both repositories have merged successfully:

A key part of the structure created is the ability to have a merge of two Git repositories (one with the Code and one with the Data) while keeping track of their commit history (and allowing pulls of new content/data).

This is achieved using the Git subtree merge strategy, which can be seen in action if you run the `$ gitk` command (note: I did a `$ git stash` first to remove a couple temporary files created by the TeamMentor engine).

The screenshot shows the gitk graphical interface displaying the commit history of a Git repository named 'TeamMentor_with_OWASP_Library'. The 'master' branch is currently checked out. The commit log shows various changes made to the repository, including updates to the OWASP Library, README files, and fix links. A commit from 'Arvind' is noted as 'Completed OWASP Content Review'. The interface shows a timeline of commits with author names and email addresses.

On the Git Commit tree shown above, the dots on the left column are from the tm_master repository (the Code) and the ones on the next column are from the tm_library repository (i.e. we have the Commits from both repositories :))

5) Push changes to the main repository:

The final step is to push the changes (of both branches) to the origin repository using:

```
$ git push -f origin master:master and  
$ git push -f origin TeamMentor_3_1:TeamMentor_3_1
```

After completion we can double-check that the master branch is now the 3.2 version:

The screenshot shows a GitHub repository page for 'TeamMentor-OWASP / Master'. The 'Code' tab is selected. The commit history section shows two entries:

- Oct 05, 2012: adding Library files by DinisCruz, authored 37 minutes ago. The commit hash is 9a33see985. A 'Browse code' link is available.
- Oct 04, 2012: adding v3.1 of the ReleaseNotes.v3.1.txt by DinisCruz, authored 18 hours ago. The commit hash is 89dbd47496. A 'Browse code' link is available.

... and the TeamMentor_3_1 branch is the 3.1 version:

The screenshot shows the GitHub commit history for the `TeamMentor_3_1` branch of the `TeamMentor-OWASP/Master` repository. The commits listed are:

- added small line about recent content update** (DinisCruz authored 4 months ago)
- updated GuidanceItems Cache** (DinisCruz authored 4 months ago)
- Merge branch 'master' of https://github.com/TeamMentor/OWASP_Library** (DinisCruz authored 4 months ago)

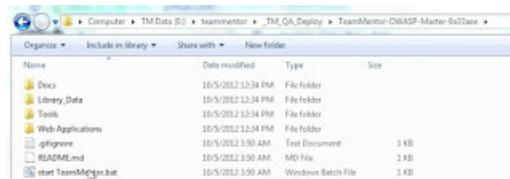
Each commit has a 'Browse code' link next to its hash.

6) Download the zip file and check that everything is okay:

As a final test, download the zip file from the repository **master branch** home page:

The screenshot shows the GitHub home page for the `TeamMentor-OWASP/Master` repository. The `master` branch is selected. In the navigation bar, there is a blue button labeled **ZIP** with a hand cursor icon pointing at it.

Extract the zip file somewhere on your disk, stop the running Cassini server (if it's still running) and double-click on the 'Start TeamMentor.bat' file:



And you will see a clean 3.2 version of TeamMentor running locally:

TeamMentor
a Security Innovation Knowledge Center

Applied Filters

- Phase = Implementation
- Technology = Java

Guidance Libraries

- OWASP
 - OWASP Top 10 - 2010
 - A01: Injection
 - A02: Cross-Site Scripting (XSS)
 - A03: Broken Authentication and Session Management
 - A04: Insecure Direct Object References
 - A05: Cross-Site Request Forgery (CSRF)
 - A06: Security Misconfiguration
 - A07: Insecure Cryptographic Storage
 - A08: Failure to Restrict URL Access
 - A09: Insufficient Transport Layer Protection
 - A10: Unvalidated Redirects and Forwards

Selected Guidance Item

All Data Passed Between Native And Java Code Is Validated

What to Check For

Verify that your application validates all data that is passed between Java code and native code.

Why

Unchecked input and/or output can lead to buffer overflows, injection based attacks such as SQL injection, cross-site scripting, etc., that exploit weaknesses in the application.

How to Check

Due to the security risk posed by the use of native code, verify that your application

Title	Technology	Phase	Type	Category
Access to the Credential Store is Limited	Java	Implementation	Checklist Item	Authentication
All Data Passed Between Native And Java Code Is Validated	Java	Implementation	Checklist Item	Input and Data Validation
All Database Input Is Validated	Java	Implementation	Checklist Item	Data Access
All Output Data Is	Java	Implementation	Checklist	Input and Data

Idea: Sync Blogger Posts with a GitHub repository

From the end of the [So if my blog account is compromised can I sue Google?](#)⁷ post comes an interesting WebService idea:

Sync Blogger posts with a GitHub repository

The idea is to backup the contents of a blogger account into a Git repository hosted by GitHub, which would give it version control and reusability.

In practice this shouldn't be that hard:

- Subscribe to RSS feed (starting with the big XML export that Blogger already provides)
- Create Git repository locally with ability to:
 - Push to GitHub
 - Download
 - Pull directly
- There needs to be some thinking on the best way to organize the files on the git repository
- It would be really cool if the files could be stored in a way that they could be consumed by other tools (like TeamMentor or <http://jekyllrb.com/>⁸)

⁷<http://diniscruz.blogspot.co.uk/2012/10/so-if-my-blog-account-is-compromised.html>

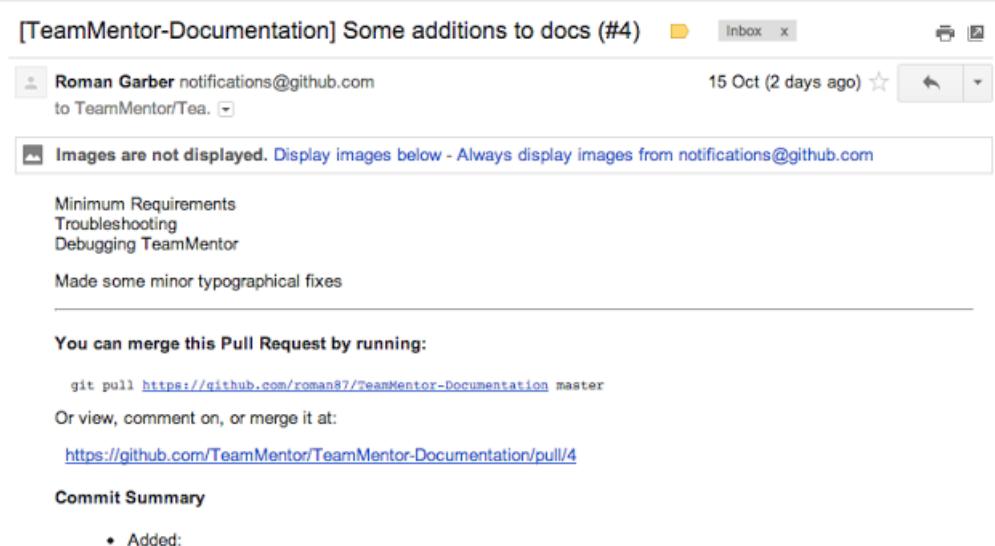
⁸<http://jekyllrb.com/>

Approving a GitHub Pull Request Workflow

I just received a GitHub's *Pull Request* from Roman for some new content that he added to TeamMentor's Documentation Site.

Here is the workflow I used to approve this request using GitHub's web-based workflow:

1) Receive GitHub email alert:



The screenshot shows an email from GitHub's inbox. The subject is "[TeamMentor-Documentation] Some additions to docs (#4)". The sender is "Roman Garber notifications@github.com" (to TeamMentor/Tea.). The email was sent on "15 Oct (2 days ago)". The message body contains a summary of changes:

Images are not displayed. Display images below - Always display images from notifications@github.com

Minimum Requirements
Troubleshooting
Debugging TeamMentor

Made some minor typographical fixes

You can merge this Pull Request by running:

```
git pull https://github.com/roman87/TeamMentor-Documentation master
```

Or view, comment on, or merge it at:

<https://github.com/TeamMentor/TeamMentor-Documentation/pull/4>

Commit Summary

- Added:

2) Go to GitHub and see the Pull Request there:

The main page (<https://github.com/TeamMentor/TeamMentor-Documentation/pull/4>)⁹ gives us a nice overview of the Pull Request:

⁹<https://github.com/TeamMentor/TeamMentor-Documentation/pull/4>

PUBLIC TeamMentor / TeamMentor-Documentation Unwatch 3 3

Code Network Pull Requests 1 Issues 1 Wiki Graphs Admin

Open roman87 wants to merge 1 commit into **TeamMentor:master** from **roman87:master** 161 #4

Discussion 1 6

roman87 opened this pull request 2 days ago **Some additions to docs**

No one is assigned - No milestone - + 146 additions - 15 deletions

Minimum Requirements
Troubleshooting
Debugging TeamMentor

Made some minor typographical fixes

1 participant

Roman87 added a commit 2 days ago
 Roman87 Added: ... 23bb660

This pull request can be automatically merged. **Merge pull request**

Here are the Commits:

Open roman87 wants to merge 1 commit into **TeamMentor:master** from **roman87:master** 161 #4

Discussion 1 6

Showing 1 unique commit by 1 author.

Oct 15, 2012 Roman87 Added: ... 23bb660

Minimum Requirements
Troubleshooting
Debugging TeamMentor

Made some minor typographical fixes

On the **Files Changed** tab we can easily see (in colours) the proposed changes.

For example, here is a change to the main TM Library xml file with a couple new articles added:

Open roman87 wants to merge 1 commit into TeamMentor:master from roman87:master 161 #4

Discussion - Commits Files Changed 6

Showing 6 changed files with 146 additions and 15 deletions. Show Diff Stats

View file @ 23bb660

```

3 ... Library_Data/XmlDatabase/TM_Libraries/TM Documentation.xml
...
4 ... @@ -4,6 +4,7 @@
5 ...     <libraryStructure>
6 ...         <view caption="Installation" id="42d43763-f674-4929-98f4-71a3345a5ec8" creationDate="0001-01-01T00:00:00">
7 ...             <items>
8 ...                 +
9 ...                     <item>00000000-0000-0000-000000410a3a</item>
10 ...                     <item>00000000-0000-0000-00000017c916</item>
11 ...                     <item>00000000-0000-0000-000000724b8</item>
12 ...                     <item>00000000-0000-0000-00000056909e</item>
...
13 ... @@ -18,6 +19,8 @@
14 ...             <item>00000000-0000-0000-0000005d8802</item>
15 ...             <item>00000000-0000-0000-00000017c916</item>
16 ...             <item>00000000-0000-0000-0000002cdf6</item>
17 ...                 +
18 ...                     <item>00000000-0000-0000-00000030106a</item>
19 ...                     <item>00000000-0000-0000-000000c4a16</item>
20 ...             </items>
21 ...         </view>
22 ...         <view caption="UI Elements" id="dbd83828-15b8-453e-ba97-261cfddde3fcc" creationDate="0001-01-01T00:00:00">
23 ...     </libraryStructure>

```

Here are a couple lines removed and some added:

```

... ...
60 ... @@ -60,10 +92,8 @@
61 ...     <li>editor : !itmeditor</li>
62 ...     <li>reader : !itmreader</li>
63 ...     </ul>
64 ...     - <b style="font-size: 10pt; ">SUPPORT INFORMATION</b>
65 ...     - <br />
66 ...     - <font size="2">For more information about TeamMentor, please see "TeamMentor Documentation" at </font>
67 ...     - <a href="http://docs.teammendor.net" style="font-size: 10pt; ">http://docs.teammendor.net</a>
68 ...     + <a href="http://docs.teammendor.net" style="font-size: 10pt; ">
69 ...     \ No newline at end of file

```

This is a new file:

```

23 [REDACTED] ...y_Data/XmlDatabase/TM_Libraries/TM Documentation/_GuidanceItems/00000000-0000-0000-0000-...
  ...
  @@ -0,0 +1,23 @@
  1 +<?xml version="1.0"?>
  2 +<TeamMentor_Article Metadata_Hash="0" Content_Hash="0">
  3 +  <Metadata>
  4 +    <Id>00000000-0000-0000-0000-00000030106a</Id>
  5 +    <Library_Id>eb39d862-f752-4d1c-ab6e-14ed697397c0</Library_Id>
  6 +    <Title>Troubleshooting</Title>
  7 +    <Category>Administration</Category>
  8 +    <Phase>NA</Phase>
  9 +    <Technology>TeamMentor</Technology>
 10 +    <Type>Documentation</Type>
 11 +    <DirectLink>Troubleshooting</DirectLink>
 12 +    <Author />
 13 +    <Priority />
 14 +    <Status />
 15 +  </Metadata>
 16 +  <Content Sanitized="false" DataType="wikitext">
 17 +    <Data><![CDATA[Unzip of the instal file requests a password=Paid content of TeamMentor is distributed via a password protected zip file. However, users installing free content can+=Password failure on unzip of the install file=First verify that you have the correct password for the paid content. If the password is correct note that this error w+</Content>
 23 </TeamMentor_Article>
\ No newline at end of file

```

These are a couple metadata changes:

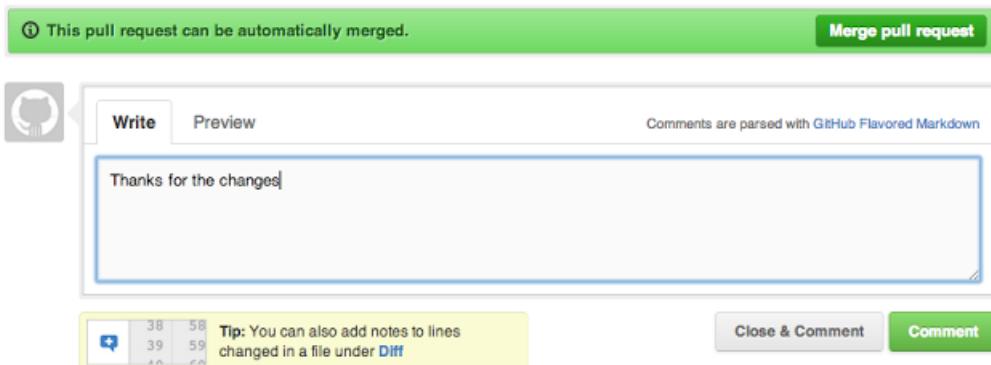
```

4 [REDACTED] ...y_Data/XmlDatabase/TM_Libraries/TM Documentation/_GuidanceItems/00000000-0000-0000-0000-...
  ...
  @@ -3,12 +3,12 @@
  3   <Metadata>
  4     <Id>00000000-0000-0000-0000-0000004866a5</Id>
  5     <Library_Id>eb39d862-f752-4d1c-ab6e-14ed697397c0</Library_Id>
  6 -    <Title>Team Mentor Plugin for Visual Studio</Title>
  6 +    <Title>TeamMentor Plugin for Visual Studio</Title>
  7     <Category>Visual Studio Plugin</Category>
  8     <Phase>NA</Phase>
  9     <Technology>Team Mentor</Technology>
 10    <Type>Documentation</Type>
 11 -    <DirectLink>Team Mentor Plugin for Visual Studio</DirectLink>
 11 +    <DirectLink>TeamMentor Plugin for Visual Studio</DirectLink>
 12    <Author />
 13    <Priority />
 14    <Status />

```

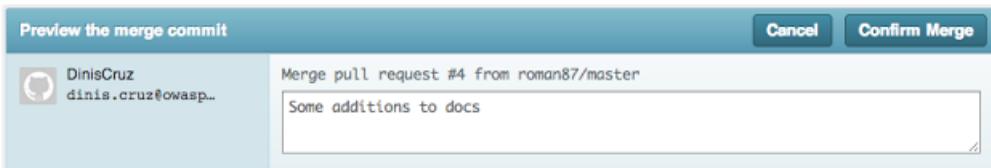
3) Approving the Pull request

Going back to the first page (the *Discussion* tab) the most important part of this whole process is the green bar that shows that this Pull Request can be merged ok



This basically means that there are no conflicts between the new changes and the current content. When this is not possible (and you get a red bar), the best thing is to do this ‘manually’ (i.e. via a git bash on your local box)

When you click on the ‘Merge Pull Request’ button you get this confirmation request:



And clicking on ‘Confirm Merge’ will do the commit and close this Pull Request:

PUBLIC  TeamMentor / TeamMentor-Documentation

[Pull Request](#) [Unwatch](#) [Star](#) 3 [Fork](#) 3

Code Network **Pull Requests 0** Issues 0 Wiki Graphs Admin

Closed DinisCruz merged 1 commit into [TeamMentor:master](#) from [roman87:master](#) less than a minute ago 161  #4

Discussion [Commits 1](#) [Files Changed 6](#)

 roman87 opened this pull request 2 days ago

Some additions to docs [Edit](#)

No one is assigned [Assign](#) No milestone [Select](#)

Minimum Requirements
Troubleshooting
Debugging TeamMentor

Made some minor typographical fixes

2 participants  

Deploying TeamMentor to AppHarbor (.NET Cloud) using Git push

Now that AppHarbor supports git publishing (i.e. they create a git repository for an ‘AppHarbor Application’), it is very easy and fast to deploy a new version of TeamMentor.

First step is to go to AppHarbor.com and create an application (for this example I’m calling it ‘Testing-AppHarbor’):

The screenshot shows the AppHarbor web interface. At the top, there's a navigation bar with links for 'Your Applications', 'How It Works', 'Pricing', 'Add-ons', and 'Support'. On the right, there's a user profile for 'DinisCruz' with a 'Sign out' option. Below the navigation, the main title is 'Applications' with a 'Full guide available here' link. A sidebar on the left lists existing applications: 'TM_Documentation' and 'TM-Top-Vulnerabilities', both owned by 'DinisCruz'. In the center, there's a form for creating a new application. The 'Name' field contains 'Testing-AppHarbor', the 'Region' dropdown is set to 'United States', and a 'CREATE NEW' button is visible. A tooltip above the 'CREATE NEW' button says 'Start by typing the name of the application'.

Once the application is created, click on the bottom-left ‘Repository URL’ button, which will copy to your clipboard the AppHarbor Git url:

The screenshot shows the 'Testing-AppHarbor' application details page. On the left, a sidebar lists various settings: 'Add-ons', 'Certificates', 'Collaborators', 'Configuration variables', 'Errors', 'Hostnames', 'Service hooks', 'Settings', and 'Subscription'. At the bottom of this sidebar are two buttons: 'REPOSITORY URL' and 'BUILD URL'. The main content area has a 'Success' message: 'Copied repository URL to clipboard'. Below this, there's a 'Getting started' section with a bulleted list: 'Configure Bitbucket to deploy to AppHarbor', 'Configure CodePlex to deploy to AppHarbor', and 'Configure GitHub to deploy to AppHarbor'. A note below says 'Not using one of those services? Deploy code using your application's built-in Git repository.' At the bottom, there's an 'Installed add-ons' section with the text 'No add-ons installed. Install add-ons from the add-on catalog'.

In this example it was: <https://DinisCruz@appharbor.com/testing-appharbor.git>

Next open an GitBash on the repository you want to push and execute:

```
$ git remote add appharbor_test https://DinisCruz@appharbor.com/testing-appharbor.git
```

```
$ git push appharbor_test master
```

After a bit (depending on your upload speed) you should get a Git push message

```
o2@WIN-FGNQ5AARJ80 /e/teammentor/_TM_Releases/TeamMentor_3.3_Dev (master)
$ git remote add appharbor_test https://DinisCruz@appharbor.com/testing-appharbor.git
o2@WIN-FGNQ5AARJ80 /e/teammentor/_TM_Releases/TeamMentor_3.3_Dev (master)
$ git push appharbor_test master
Password for 'https://DinisCruz@appharbor.com':
To https://DinisCruz@appharbor.com/testing-appharbor.git
 * [new branch]      master -> master
```

Back in the AppHarbor website, the application page should look like this:

The screenshot shows the AppHarbor application management interface. On the left, there's a sidebar with a dark blue header containing the application name 'Testing-AppHarbor'. Below the header, the sidebar lists several settings: Add-ons, Certificates, Collaborators, Configuration variables, Errors, Hostnames, Service hooks, Settings, and Subscription. At the bottom of the sidebar are two buttons: 'REPOSITORY URL' and 'BUILD URL'. The main content area has a light gray background. At the top, it says 'Build status' and 'Monitor build system status'. Below this is a table with columns: STATUS, RECEIVED, DEPLOYED, COMMIT ID, and COMMIT MESSAGE. There is one row in the table with the following data: a blue circular icon with a white 'C' (animating), the date '10/17/12 3:23 PM', the commit ID 'cb4366d...', and the commit message 'set version to TM 3.2.2'. Below the table, under the heading 'Installed add-ons', it says 'No add-ons installed. Install add-ons from the add-on catalog'.

You can click on the icon under the 'Status' column (the one animating) to see the current status of the build (note that sometimes AppHarbor takes a couple minutes to trigger the compilation process)

The screenshot shows the AppHarbor build details page for build cb4366... The left sidebar contains navigation links: Add-ons, Certificates, Collaborators, Configuration variables, Errors, Hostnames, Service hooks, Settings, and Subscription. Below these are two buttons: REPOSITORY URL and BUILD URL. The main content area includes fields for Branch name (master), Commit ID (cb4366da9b0946ba05e3ae935d9d5bd125c8ad21), and Commit message (set version to TM 3.2.2). A Log section lists the following events:

- 10/17/12 3:23 PM: Received notification, queuing build
- 10/17/12 3:23 PM: Downloading source
- 10/17/12 3:23 PM: Downloaded source in 7.82 seconds
- 10/17/12 3:23 PM: Starting build
- 10/17/12 3:23 PM: 0 warnings
- 10/17/12 3:23 PM: Build completed in 5.61 seconds Details
- 10/17/12 3:23 PM: Starting website precompilation
- 10/17/12 3:23 PM: Precompilation completed in 6.39 seconds
- 10/17/12 3:23 PM: Starting tests
- 10/17/12 3:23 PM: Tests completed in 2.64 seconds

The ‘Details’ link can be used to see the MSBuild compilation log (very useful when the compilation fails)

The screenshot shows a browser window displaying the MSBuild compilation log for build cb4366... The log output is as follows:

```

Build started 10/17/2012 3:23:39 PM.
1>Project "D:\temp\xeva4ws.033\input\Web Applications\TM_Website.sln" on node 1 (default targets).
1>ValidateSolutionConfiguration:
   Building solution configuration "Release|Mixed Platforms".
1>Project "D:\temp\xeva4ws.033\input\Web Applications\TM_Website.sln" (1) is building "D:\temp\xeva4ws.033\input\Web Applications\TM_Website\TM_Website.csproj" (2) on node 1 (default targets).
2>PrepareForBuild:
   Creating directory "obj\Release".
1>Project "D:\temp\xeva4ws.033\input\Web Applications\TM_Website.sln" (1) is building "D:\temp\xeva4ws.033\input\Web Applications\TeamMentor.CoreLib\TeamMentor.CoreLib.csproj" (3) on node 2 (default targets).
3>PrepareForBuild:
   Creating directory "obj\Release".
CoreCompile:
   C:\Windows\Microsoft.NET\Framework\v4.0.30319\Csc.exe /noconfig /nowarn:1701,1702 /nostdlib+ /errorreport:prompt /warn:4 /define:TRACE /highentropyva- /reference:"D:\temp\xeva4ws.033\input\Web Applications\packages\AntiXSS.4.2.1\lib\net40\AntiXssLibrary.dll"
   /reference:"D:\temp\xeva4ws.033\input\Web Applications\packages\AntiXSS.4.2.1\lib\net40\HtmlSanitizationLibrary.dll"
   /reference:"D:\temp\xeva4ws.033\input\Web Applications\packages\SharpZipLib.0.86.0\lib\20\ICSharpCode.SharpZipLib.dll" /reference:"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\mscorlib.dll" /reference:"D:\temp\xeva4ws.033\input\Web Applications\packages\FluentSharp.CoreLib.4.3.2.0\lib\net35\O2_FluentSharp_CoreLib.dll" /reference:"D:\temp\xeva4ws.033\input\Web Applications\packages\O2_Platform.Misc.Microsoft_MPL_Libs.4.1.0.0\lib\net35\O2_Misc.Microsoft_MPL_Libs.dll" /reference:"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\System.Data.DataVisualization.dll" /reference:"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\System.Data.dll" /reference:"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\System.DirectoryServices.AccountManagement.dll" /reference:"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\System.dll" /reference:"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\System.Web.dll" /reference:"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.0\System.Web.Extensions.dll" /reference:"C:\Program Files (x86)\Reference

```

Once the build and deployment is done, if you go back to the Application page you should see that your build is now Active

STATUS	RECEIVED	DEPLOYED	COMMIT ID	COMMIT MESSAGE
	10/17/12 3:23 PM	10/17/12 3:24 PM	cb4366...	set version to TM 3.2.2

And clicking on ‘Go to your application’ will take you to the website you just created

By default there are no libraries installed from a [TeamMentor/Master¹⁰](#) clone/copy, but this is easily solved by:

- logging in as **Admin**,
- going to the **Control Panel**
- choosing the **“Advanced Admin Tools”** option
- using the **Install/Upload Libraries** tool.

For example, click on the ‘**Top 20 Vulnerabilities**’ link to add the Library hosted at GitHub’s [TeamMentor/Library_Top_Vulnerabilities¹¹](#) (TeamMentor engine will go to that repository, download the ZIP file and install it)

¹⁰<http://github.com/teammentor/master>

¹¹https://github.com/TeamMentor/Library_Top_Vulnerabilities

User Management

- My Account
- Manage Users
- Create Multiple Users

BackEnd

- Set Library Path
- Admin Tasks
- TeamMentor Web Services

Developer Utils

- Install/Upload Libraries
- Firebug Lite
- Web Editor
- GitHub Sync

- TeamMentor QUnit Tests

- UIAutomation (Admin)

- UIAutomation (User)

Misc links:

- Login
- Logout
- Advanced Admin Tools

Navigation

- Open Main Page

Install Library

Library Zip files uploaded to this server (click to install)

Current Libraries Zip Folder(server side):
D:\websites\ad\2be0419528\0x0001_PublishedWebsites\TM_Website\App_Data\Library_Data\Xml\Database\TM_Library_UploadedFiles

Note that instalation from Zip will override existing libraries with the same name. [refresh list](#)

Install Library from File or Url

URLs for Libraries to install: [OWASP Top 20 Vulnerabilities](#) [TM Documentation](#) [Java \(needs unzip pwd\)](#) [.Net 4.0 \(needs unzip pwd\)](#)

File (server-side) or Url: https://github.com/TeamMentor/Library_Top_Vulnerab

Unzip Password:

[Install](#)

> Library installed was successful

Upload Library to server

[Upload a file](#)

Once that is done, click on ‘Open Main Page’ to go back to the main TM Gui where the ‘Top 20 Vulnerabilities’ Library is now installed:

Applied Filters

Guidance Libraries

- Top Vulnerabilities
 - 01 XSS
 - 02 SQLI
 - 03 CSRF
 - 04 Buffer Overflow
 - 05 Format String
 - 06 XML Injection
 - 07 SSL Misconfiguration
 - 08 Session Hijacking
 - 09 Weak Cryptography
 - 10 Information Disclosure
 - 11 Direct Request
 - 12 Forceful Browsing
 - 13 URL Redirection
 - 14 Unsecure Cookies
 - 15 Poor Password Storage
 - 16 Authentication Issues
 - 17 Key Mismangement
 - 18 Single Sign-on Issues
 - 19 Misc Command Injection
 - 20 Access Control Issues

Selected Guidance Item

Allocate Enough Buffer Space for Copying Data

What to Do

Only copy the amount of data that can fit in the array or string. Make sure that the array or string is large enough to fit all the data, including the possible trailing null.

Why

Copying more data than an array or string can fit results in a buffer overflow.

When

Allocate enough buffer space for copying data, when data is being copied to an array or a string.

How

Title	Technology	Phase	Type	Category
Allocate Enough Buffer Space for Copying Data	Native Code	Implementation	Guideline	Memory Management
Allow Only Trusted Hosts to Connect to the Database Server	ASP.NET	Implementation	Principle	Data Access
Centralize Authentication Controls	Any	Design	Principle	Authentication
Change Session IDs				Session

Auto deploy on GitHub Commit

So far we talked about how to push an git repository into AppHarbor from your local disk, but that is not the only way you can do it.

AppHarbor also supports [GitHub Service Hooks](#)¹² which can be configured via the GitHub's repository admin panel.

For example, here is how I use AppHarbor to create a new deployment every-time I do a Git Push into the main TeamMentor/Master repository (which is very useful for QA and Testing)

The screenshot shows the GitHub Repository Administration interface for the 'TeamMentor / Master' repository. The left sidebar has a 'Service Hooks' section selected. The main area lists 'AVAILABLE SERVICE HOOKS' including WebHook URLs (0), ActiveCollab, Acunote, AgileBench, AgileZen, AMQP, Apolo, AppHarbor (selected), Backlog, Bamboo, BasecampClassic, and Basecamp. On the right, there are fields for 'Application Slug' (teammentor-3-2) and 'Token' (XXXXXXXXXXXXRuK%2f1dvkhQldVtcfIZMpIG%2bMmIioGM!), both of which are highlighted with blue boxes. A checked checkbox labeled 'Active' is also present. At the bottom right are 'Test Hook' and 'Update Settings' buttons. A note at the bottom states: 'AppHarbor is a .NET Platform as a Service. Use this service to automatically trigger build, test run and deployment of your AppHarbor application when you push to GitHub.'

¹²<https://github.com/github/github-services>

The need to create forks/clones for TeamMentor.net website

(Here is an email I sent earlier today at SI, that covers a number of interesting challenges that we're now having with TeamMentor, and how Git/GitHub can help)

One of the scenarios/problems that is starting to happen is '**how to manage the specific requirements for deployed sites like Teammentor.net who need custom changes?**'

So here is a description of 'the problem':

- a) there is a master version of the code: [`https://github.com/TeamMentor/Master`](https://github.com/TeamMentor/Master)¹³
- b) there is a master version of the content: [`https://github.com/TeamMentor/Library_SI`](https://github.com/TeamMentor/Library_SI)¹⁴
- c) there is a master version of the code+content (which is a 'virtual copy' of the two above): [`https://github.com/TeamMentor/TeamMentor_SI_Library`](https://github.com/TeamMentor/TeamMentor_SI_Library)¹⁵ (think of this as a copy of the code and master repositories in a way that keeps the git connections and commit history)
- d) there is a website that is based on the code + content version (i.e. [`http://teammentor.net`](http://teammentor.net)¹⁶) , BUT has a number of specific requirements.
 - the content should not be publicly available (not TM default), ie require account creation
 - specific SI requirements (see this list: [`https://github.com/TeamMentor/Master/issues?milestone=3&state=open`](https://github.com/TeamMentor/Master/issues?milestone=3&state=open)
- e) how to deal with changes to [`http://teammentor.net`](http://teammentor.net)¹⁸ , that are only relevant to that site (i.e will not be propagated to the main Code+Content repo)
 - interestingly in the case of TeamMentor.net site, (at least at the moment), the only changes will be on the Code (where the content is the same as the one in [`https://github.com/TeamMentor/TeamMentor_SI`](https://github.com/TeamMentor/TeamMentor_SI)¹⁹)

¹³[`https://github.com/TeamMentor/Master`](https://github.com/TeamMentor/Master)

¹⁴[`https://github.com/TeamMentor/Library_SI`](https://github.com/TeamMentor/Library_SI)

¹⁵[`https://github.com/TeamMentor/TeamMentor_SI_Library`](https://github.com/TeamMentor/TeamMentor_SI_Library)

¹⁶[`http://teammentor.net/`](http://teammentor.net/)

¹⁷[`https://github.com/TeamMentor/Master/issues?milestone=3&page=1&state=open`](https://github.com/TeamMentor/Master/issues?milestone=3&page=1&state=open)

¹⁸[`http://teammentor.net/`](http://teammentor.net/)

¹⁹[`https://github.com/TeamMentor/Library_SI`](https://github.com/TeamMentor/Library_SI)

- but I can see how as our content generation capabilities improves (and we start to have fresh and ‘current/recent-events’ articles) we might want to push those into [teammentor.net²⁰](http://teammentor.net) (since they will be the best advertisement for TM that we could ever get)
- f) since [teammentor.net²¹](http://teammentor.net) is a ‘read-only’ website, there is also an argument (from a security point of view) that that site should not have the advanced editing capabilities that TeamMentor has (this is also a feature that I can see customers wanting)

So what is the solution?

At the moment, the solution that I see is to:

- create a (private) fork of [https://github.com/TeamMentor/TeamMentor_SI_Library²²](https://github.com/TeamMentor/TeamMentor_SI_Library) (the Code+Content) repository at TMClients organisation (technically it will be a clone not a fork).
- apply the Code changes (specific to SI) in that fork
- use that fork as the source repository of [teammentor.net²³](http://teammentor.net)

Any other ideas on how to deal with this?

Note that we have the exact same issue (or a variation of this) on:

- [https://owasp.teammentor.net²⁴](https://owasp.teammentor.net) (which will have for example ‘Google analytics’ tracking by default and ‘SSL only’ enabled)
- [https://docs.teammentor.net²⁵](https://docs.teammentor.net) (used to host the tm documentations)
- [https://download.teammentor.net²⁶](https://download.teammentor.net) (tm site soon to be created to host the Marketing pages (for example the current [https://docs.teammentor.net/xml/Eval²⁷](https://docs.teammentor.net/xml/Eval) and [https://docs.teammentor.net/xml/Customer²⁸](https://docs.teammentor.net/xml/Customer))

²⁰<http://teammentor.net/>

²¹<http://teammentor.net/>

²²https://github.com/TeamMentor/TeamMentor_SI_Library

²³<http://teammentor.net/>

²⁴[https://owasp.teammentor.net/](https://owasp.teammentor.net)

²⁵[https://docs.teammentor.net/](https://docs.teammentor.net)

²⁶[https://download.teammentor.net/](https://download.teammentor.net)

²⁷<https://docs.teammentor.net/xml/Eval>

²⁸<https://docs.teammentor.net/xml/Customer>

- <https://teammentor.teammentor.net>²⁹ (also called tm4tm, which is a site that will host all technical and non-tm-user documentation/info about TeamMentor (for example https://docs.teammentor.net/xsl/Table_of_Contents³⁰ will move to this tm4tm site)
- a client that wants to make code changes (for example on TM GUI), most likely another security company (a Partner) or a security department inside a bigger company

²⁹<https://teammentor.teammentor.net/>

³⁰https://docs.teammentor.net/xsl/Table_of_Contents

Using Git Branches to deal with the multi-config variations of TeamMentor

Here is an interesting problem that affects TeamMentor (TM) and just about every other app:
“How to deal with the specialized versions of an application that are created via Config changes”

Keeping this simple, and only dealing with the config changes that can be made by modifying the [TmConfig.config³¹](#) file, TM already (today) has the following scenarios to support:

- Default install (with default settings)
- Anonymous users cannot see the content (with a variation where anonymous users cannot see the Library View (not done via config change))
- Redirect all Http traffic toHttps (i.e. SSL Redirect)
- Windows Authentication enabled/disabled
- SSO enabled/disabled
- Change location of TM Libraries
- Enforce HTML Sanitisation on Article content
- Change default admin pwd

There are also other scenarios that I’m sure TM will need to support very soon:

- Read-only version of TM
- ‘Secure / lock-down’ version(s) of TM
- OAuth integration
- Support for 3rd party data sources (like the PoC done where wikipedia, msdn and owasp content is consumed by TM natively)

I was thinking about the ways to solve this, and here are a couple options:

1. Rely on user documentation and pass the responsibility to *‘apply the changes correctly’* to the customers/users (I don’t like this solution, although is what most vendors do (including SI))

³¹https://github.com/TeamMentor/Master/blob/master/Web%20Applications/TM_Website/https://github.com/TeamMentor/Master/blob/master/Web%20Applications/TM_Website/TmConfig.config

2. Create forks that apply the required changes only to those repositories (this is what we currently do in the forks we maintain)
3. Make changes on specialized Git Branches, and use Git Checkout to enable them (hum....)

As you can see by my comments:

- I really don't like option #1,
- option #2 is what we currently do on some cases (and does have the side effect of fork-explosion), and
- option #3 is an idea I have been thinking for a while, and the more I think about it, the more I like it.

Basically the idea would be to use Git Branches to track/apply those config changes.

Currently we use Git Branches to hold references to past TM versions (like we do at the moment in the 'master TM repositories'). Note that the Git Branches used for special dev tests on Dev forks would not be affected by this.

What I really like about this idea is:

- it would put the responsibility to create those 'variations' in the most capable hands (i.e. the ones who know the code best)
- it would allow for a strong QA cycles and for much better support for those scenarios
- it would make life easier for customers/users
- it would provide a scalable way to support complex configuration changes (i.e. scenarios that require more than a couple 'settings-changes')

What do you think?

Any other ideas on how to deal with this issue?

Handling content changes made on hosted site created by Git clone (with auto Git commits and pushes)

Here is the next interesting TeamMentor (TM) and GitHub problem to solve:

“How to deal with content changes made using TM’s online editing capabilities”

Here is the ‘Problem description’:

- TM is published from a Git Repository into a Server (let’s call it from **REPOSITORY A** to **SITE A**)
- There are two deployment modes that already work well:
- Publish to cloud (Azure or AppHarbor): takes about 2 to 5 minutes
- Publish to EC2 server: takes between 10 to 30 minutes and includes custom DNS and IIS set-up/deployment
- In either mode the idea is that **REPOSITORY A** is the master version of the Code+Data
- This means that if we needed to rebuild that site, we could (ie. should) be able do it in minutes
- Upgrades and patches are made via a simple git pull (which gets the latest version from the **REPOSITORY A**) and no git merge activity should be needed
- But what happens when there is a content change on **SITE A**’s files?
- An automated solution is needed, since the option of ‘*RDPing into the server to do the commits/push*’ or ‘*trying to do the commits/push via TM’s GitHub interface*’, not only don’t scale but are as dangerous as relying on manual backups.

Here is what I have in mind:

- TM detects if git support exists on the deployed server (i.e git.exe is available) and:
- Git checkout (the deployed branch) into a special ‘live_server’ branch
- Auto git commit on every TM Content save (or creation) with the commit message being a mix of: Current user, its IP, the date and time and the file affected
- (if configured) auto-push the change to an GitHub repository
- This could be done by configuring an SSH key on the server, or by hardcoding the GitHub credentials into the ‘Git remote’ value
- If git.exe is not available, then these commits and pushes will need to be done manually (by zipping the whole content folder and moving it into a location with git.exe available)

- In terms of the repository to push, I think that we shouldn't push directly to the original **REPOSITORY A** (the one used to create **SITE A**), but we should push it to a fork/clone of **REPOSITORY A** which could be used as a staging location (one where Pull Requests into **REPOSITORY A** would be made)
- Of course that we could push into **REPOSITORY A** directly, but that would expose an account with git push privileges to an important repository, and could create a scenario where unauthorized changes were made into one a production repositories (also note that with git push privileges, it is possible to completely remove all history and commits from a repository (in effect deleting all information)).

What I like about this solution (the auto-commit with option to auto-push) is that it will provide TeamMentor with a state-of-the-art version control solution (at article level).

Every single change would be logged, and although this will most likely make that branch completely unreadable by humans we will be able to have really powerful (and cool) per-article version history (ie. for each file, see its complete change log, including 'who did the change').

Note that it is possible to combine a number of commits into one (not in GitHub, but in git.exe) so I think that for cases where a number of files were changed, we might want to consolidate them into larger commits (specially when pushing those changes to 'user consumable' repositories)

What do you think?

Any better ideas?

Going back in time using Git's checkout

I'm still amazed at Git's speed in moving back and forwards in time. For example I was trying to find a particular GUI that we created for TeamMentor and was able to use git checkout to look at previous versions:

`git checkout master (25 Oct 2012 TM 3.2)`

The screenshot shows a web browser displaying the TeamMentor application at `127.0.0.1:12120/teamMentor`. The interface includes a header with the TeamMentor logo and navigation links for 'Login', 'Sign Up', 'TM 3.2', and 'About'. On the left, there are 'Applied Filters' and 'Guidance Libraries' sections, with 'OWASP' and 'OWASP Top 10 - 2010' selected. The main content area shows a search bar and three filter dropdowns: 'Technology' (Any, ASP.NET 3.5, Java), 'Phase' (Any, Deployment, Design, Implementation), and 'Type' (Attachment, Checklist Item, Guide, How To, Principle). A table below lists 174 items, with the first item highlighted: 'A Secure Key Storage Location Is Used' (Title), 'ASP.NET 3.5' (Technology), 'Implementation' (Phase), 'Checklist Item' (Type). To the right, a detailed view of this item is shown, including sections for 'What to Check For' (ensure application keys are stored in a well-defined location) and 'Why' (when found with corruption).

`git checkout dd867bfb4b9519c3b9c6ddfe2c0f9b1f6720f162 (4th Sep 2012: TM 3.2 RC1) :`

The screenshot shows a web browser window for the 'TeamMentor' application, version TM 3.2 RC1. The URL is 127.0.0.1:12115/html_pages/Gui/TeamMentor.html. The interface includes a header with the TeamMentor logo, a search bar, and links for Login, Sign Up, and About. On the left, there's a sidebar titled 'Applied Filters' with a red 'X' button. Below it is a section for 'Guidance Libraries'. The main content area has tabs for 'Technology' and 'Phase', with a search bar above them. A message says 'Showing 0 items (out of 0)'. Below this is a table header with columns: Title, Technology, Phase, Type, and Category. A message at the bottom of the table area says 'No data available in table'. To the right, a large panel is titled 'Selected Guidance Item' with a link icon. Below it, a message says 'No article available.'

git checkout 890caa053fee04bf0b7139787e0ee6100963771 (23rd Jan 2012: TM 3.0) :

The screenshot shows a web browser window for the 'TeamMentor' application at the URL 127.0.0.1:12345/html_pages/Gui/TeamMentor.html. The page title is 'TeamMentor - a Security Innovation eKnowledge Product'. In the top right corner, there are links for 'Login', 'Sign Up', 'TM 3.0 (23 Jan)', and 'About'. On the left, there's a sidebar with 'Applied Filters' and 'Guidance Libraries'. The main content area has two tabs: 'Technology' and 'Phase'. Below them, a search bar and filter buttons for Title, Technology, Phase, Type, and Category are visible. A message 'Showing 0 items' and 'No data available in table' is displayed. To the right, a large box titled 'Selected Guidance Item' contains the message 'Login Required' with the sub-instruction 'You must be logged in to view this page. Please [Login](#) or [Sign Up](#)'. The overall interface is clean with a light blue and white color scheme.

git checkout 557177691139bf2385973b45bf39508042a11621 (18th Jan 2012: TM 3.0 RC9) :

This screenshot is identical to the one above, showing the TeamMentor application at the same URL and version. The only difference is the date and version information in the top right corner, which now reads 'TM 3.0 RC9 (18 Jan)'. The rest of the interface, including the sidebar, tabs, search filters, and the 'Selected Guidance Item' message, remains the same.

Looking at these images, I thought of a cool script to write (here it is in pseudocode):

```
1 foreach id in avaialble_checkout_
2 {
3     git checkout id
4     {
5         start webserver
6         open default page in browser (if possible 'add a library if not there')
7         take screenshot (only store unique values and if possible 'add watermark with vers\
8 ion and date')
9         close web server
10    }
11 }
12 create animation from screenshots taken _
```

It's all doable with O2's APIs (I just don't have the time today) :)

Adding Tags to TeamMentor Master repository

With 3.2 out, its time to add some [Git Tags³²](#) to the main [TeamMentor/Master³³](#) repository (which at the moment has none):

The screenshot shows a GitHub repository page for 'TeamMentor / Master'. At the top, there are buttons for 'Pull Request', 'Unwatch', 'Star', 'Fork', and 'Admin'. Below that is a navigation bar with tabs: 'Code' (which is selected), 'Network', 'Pull Requests 0', 'Issues 84', 'Wiki', 'Graphs', and 'Admin'. Underneath the navigation bar, there are links for 'Files', 'Commits', 'Branches 4', 'Tags' (which is selected), and 'Downloads 6'. The main content area displays the message: 'There aren't any tags in this repository. You can create tags by using git tag.'

In a local Git Bash of this repository, we can create a tag using `$ git tag -a v3.2 -m '3.2 Release'`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ git tag -a v3.2 -m '3.2 Release'
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ git tag
v3.2
```

Next we push that tag into GitHub using `$ git push tm_master v3.2`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ git push tm_master v3.2
Counting objects: 1, done.
Writing objects: 100% (1/1), 162 bytes, done.
Total 1 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Master.git
 * [new tag]           v3.2 -> v3.2
```

And if we look back in GitHub's Tag page, we will see that our v3.2 tag is in there:

³²<http://git-scm.com/book/en/Git-Basics-Tagging>

³³<https://github.com/TeamMentor/Master/>

This screenshot shows a GitHub repository page for 'TeamMentor / Master'. At the top, there are buttons for 'Pull Request', 'Unwatch', 'Star', 'Fork', and 'Admin'. Below the header, there are tabs for 'Code', 'Network', 'Pull Requests 0', 'Issues 84', 'Wiki', 'Graphs', and 'Admin'. Under the 'Code' tab, there are links for 'Files', 'Commits', 'Branches 4', 'Tags 1', and 'Downloads 5'. A section titled '1 tag' is shown, containing a single entry: 'v3.2.zip — 3.2 Release' with a commit hash '4aae621e6b' and a note 'Uploaded 9 minutes ago'.

At the moment we are keeping track of the previous versions using Git Branches (but I think that tags will do a better job)

This screenshot shows a GitHub repository page for 'TeamMentor / Master'. The interface is similar to the previous one, with tabs for 'Code', 'Network', 'Pull Requests 0', 'Issues 84', 'Wiki', 'Graphs', and 'Admin'. The 'Branches' tab is selected, showing four branches: 'master', '3.3_Release', '3.1_Release', and '2.1.1'. The 'master' branch is highlighted. A legend indicates 'Recently Active' (red), 'Stale' (grey), and 'Base branch' (dark grey). Comparison buttons ('Compare') are provided for each branch.

For example here is 3.1 release (with the *f71b016241* id)

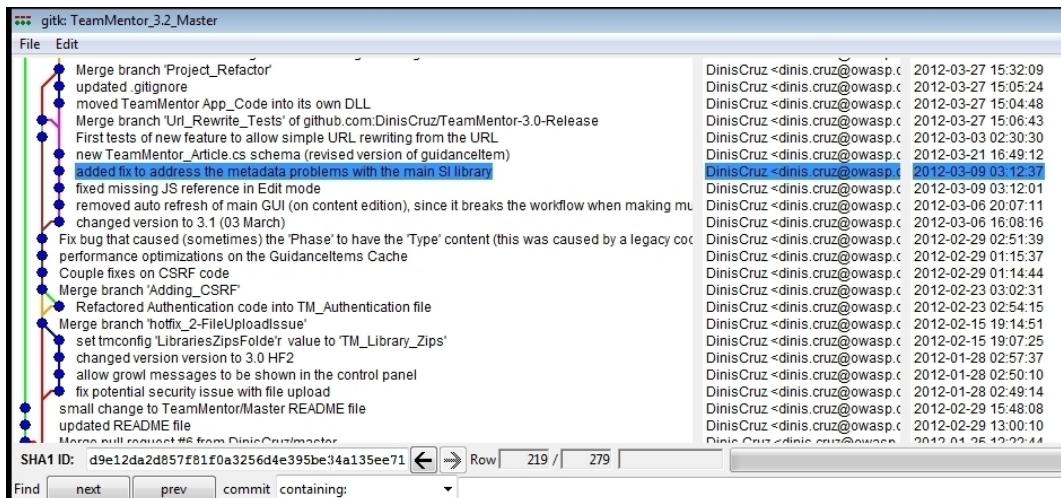
This screenshot shows a GitHub repository page for 'TeamMentor / Master'. The 'Commits' tab is selected. The commit history for the '3.1_Release' branch is displayed, starting with 'Merge pull request #8 from DinisCruz/master' (commit f71b016241, authored 8 months ago) and followed by two other commits: 'Changed version in README to 3.1' (commit 05be6efbdb, authored 8 months ago) and 'changed version to 3.1 (03 March)' (commit ab77ba07a5, authored 8 months ago). Each commit has a 'Browse code' link next to it.

We can use this ID value to create the 3.1 tag

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ git tag -a v3.1 -m '3.1 Release' f71b016241
```

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ git push tm_master v3.1
Counting objects: 1, done.
Writing objects: 100% (1/1), 162 bytes, done.
Total 1 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Master.git
 * [new tag]           v3.1 -> v3.1
```

Use `gitk` to find the SHA1 ID of the 3.0 release



Which we use to create the 3.0 tag:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ git tag -a v3.0 -m '3.0 Release' d9e12da2d857f81f0a3256d4e395be34a135ee71

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ gitk

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_3.2_Master (master)
$ git push tm_master v3.0
Counting objects: 1, done.
Writing objects: 100% (1/1), 164 bytes, done.
Total 1 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Master.git
 * [new tag]           v3.0 -> v3.0
```

After pushing to GitHub, the Tag page looks like this:

The screenshot shows a GitHub repository page for 'TeamMentor / Master'. At the top, there are buttons for 'Pull Request', 'Unwatch', 'Star' (with 4 notifications), 'Fork' (with 8 notifications), and 'Admin'. Below the header, there are tabs for 'Code', 'Network', 'Pull Requests 0', 'Issues 84', 'Wiki', 'Graphs', and 'Admin'. Under 'Code', there are links for 'Files', 'Commits', 'Branches 4', 'Tags 3', and 'Downloads 5'. The 'Tags' section displays three entries:

- v3.2.zip — 3.2 Release
4aae621e6b · Uploaded 18 minutes ago
- v3.1.zip — 3.1 Release ZIP TAR.GZ
f71b016241 · Uploaded 5 minutes ago
- v3.0.zip — 3.0 Release
d9e12da2d8 · Uploaded a minute ago

What is really cool about these Git Tags is that they also provide a nice location to download a particular release :)

Creating the final TeamMentor with SI Library repository via multiple Git pulls and pushes

This is going to be a long one, so if you are interested in seeing Git and GitHub in action in a real-world application, grab a coffee/tee/beer and read on :)

This is the scenario at the start:

- We need to create the final TeamMentor 3.2.3 package for release
- The *TeamMentor_SI_Library* repository is at version 3.2
- The *TeamMentor/Master* repository is at version 3.2.3
- The *Library_SI* has a couple content changes since the last pull
- We need to do a pull of both *TeamMentor/Master* and *Library_SI* into *TeamMentor_SI_Library* and push the final result to GitHub

For background information on the current TeamMentor git architecture, take a look at:

- [Git and GitHub commands to create and deploy new version of TeamMentor³⁴](#) (this explains the subtree merge strategy, which is the reason why the multiple git pull used below actually work)
- [New GitHub structure for the multiple TeamMentor development and deployment scenarios³⁵](#)
- [The need to create forks/clones for TeamMentor.net website³⁶](#)

It all starts with a pull request from Roman the last content changes to be added to *Library_SI*

The screenshot shows a GitHub repository interface for the 'TeamMentor' organization. The 'Pull Requests' tab is active, displaying 1 open request. The request is from 'roman87' titled 'Removed the date from OWASP Top 10 views'. The pull request was submitted a day ago and updated a day ago. There are 0 comments. The repository navigation bar includes 'News Feed', 'Pull Requests', 'Issues', and 'Teams'. On the left, there's a sidebar with repository filters ('Private', 'Public') and a search bar ('Find a repository...'). Below the sidebar, there's a list of repositories: 'TeamMentor/Library_SI' (1), 'TeamMentor/OWASP_Library' (1), 'TeamMentor/TeamMentor_Visua...' (0), and 'TeamMentor/TeamMentor-Docu' (0).

³⁴<http://diniscruz.blogspot.co.uk/2012/10/git-and-github-commands-to-create-and.html>

³⁵<http://diniscruz.blogspot.co.uk/2012/10/new-github-structure-for-multiple.html>

³⁶<http://diniscruz.blogspot.co.uk/2012/10/the-need-to-create-forksclones-for.html>

Which I opened

PRIVATE TeamMentor / Library_SI

Pull Requests 1 Issues 1 Wiki Graphs Admin

Open roman87 wants to merge 1 commit into `TeamMentor:master` from `roman87:master`

Discussion 0 Commits 1 Files Changed 4

roman87 opened this pull request a day ago

Removed the date from OWASP Top 10 views

No one is assigned No milestone

No description given.

1 participant

Quickly reviewed the changes

PRIVATE roman87 / Library_SI forked from TeamMentor/Library_SI

Code Network Pull Requests 0 Wiki Graphs Admin

Files Commits Branches 1 Tags Downloads Search source code...

Changed OWASP Top 10 2010 heading to OWASP Top 10 (removing 2010 date)

Roman87 authored a day ago 1 parent `d766b6ed4f` commit `86ee6654438f3ed1cccd4138abe441ec269862abd`

Showing 4 changed files with 4 additions and 4 deletions.

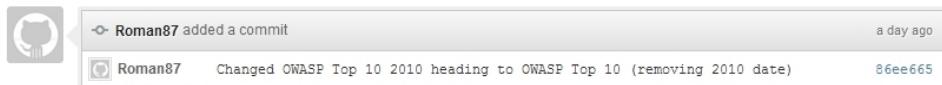
File	Changes	View file @ 86ee665
.NET 3.5/.NET 3.5.xml	2 additions, 2 deletions	View file @ 86ee665
.NET 4.0/.NET 4.0.xml	2 additions, 2 deletions	View file @ 86ee665

```

@@ -1419,7 +1419,7 @@
     </items>
   </view>
</folder>
- <folder caption="OWASP Top 10_2010" folderId="e570fb50-cab2-44dc-b01f-6287fd06da08">
+ <folder caption="OWASP Top 10" folderId="e570fb50-cab2-44dc-b01f-6287fd06da08">
    <columns>
      <column source="Type" caption="Type" sort="Ascending" visible="true" width="111" position="1" />
      <column source="Topic" caption="Topic" sort="Ascending" visible="true" width="111" position="2" />

```

And Confirmed Merge



You can add more commits to this pull request by pushing to the `master` branch on `roman87/Library_SI`



This will close the Pull Request

PRIVATE TeamMentor / Library_SI Pull Request Unwatch Star Fork Admin
Code Network Pull Requests 0 Issues 0 Wiki Graphs Admin
Closed DinisCruz merged 1 commit into TeamMentor:master from roman87:master less than a minute ago 8 #1
Discussion Commits 1 Files Changed 4
roman87 opened this pull request a day ago
Removed the date from OWASP Top 10 views
Edit
No one is assigned No milestone
No description given.



With changes now part of the main *Library_SI* repository

PRIVATE TeamMentor / Library_SI

Pull Request Unwatch Star Fork 1

Code Network Pull Requests 0 Issues 0 Wiki Graphs Admin

branch: master Files Commits Branches 1 Tags Downloads Search source code...

Keyboard shortcuts available

Library_SI / Commit History

Oct 27, 2012

Merge pull request #1 from roman87/master ... 488ba49d6a +
DinisCruz authored just now Browse code ↗

Oct 26, 2012

Changed OWASP Top 10 2010 heading to OWASP Top 10 (removing 2010 date) 86ee665443 +
Roman87 authored a day ago Browse code ↗

Oct 04, 2012

Adding Android and IOS library files d766b6ed4f +
DinisCruz authored 23 days ago Browse code ↗

For reference here is the network graph of *Library_SI*

PRIVATE TeamMentor / Library_SI

Pull Request Unwatch Star Fork 1

Code Network Pull Requests 0 Issues 0 Wiki Graphs Admin

Graph Members

The Library_SI network graph

All branches in the network using TeamMentor/Library_SI as the reference point. Read our blog post about how it works.

Keyboard shortcuts available

Sep 10 18 19 25 27 30 28 Oct 4 26 27

TeamMentor

Next we move into the TeamMentor_SI_Library:

TeamMentor / TeamMentor_SI_Library

Code Network Pull Requests 0 Issues 0 Wiki Graphs Admin

branch: master Files Commits Branches 2 Tags Downloads Search source code... 🔍

TeamMentor_SI_Library / Commit History ↗

Oct 12, 2012

💻 Changed TmConfig.config to require login to see the content. b00ed106b4 ↗
Roman87 authored 15 days ago Browse code ↗

Oct 05, 2012

💻 adding "Docs/Util - Git commands used to create this version.txt" 92ad3b7656 ↗
DinisCruz authored 22 days ago Browse code ↗

💻 Minor changes for TM+SI version (different Cassini port and README.md) 68efb0e1be ↗
DinisCruz authored 22 days ago Browse code ↗

💻 adding Library files 6373d9714a ↗
DinisCruz authored 22 days ago Browse code ↗

Opening the local copy of the *TeamMentor_SI_Library*, as the log list shows (below) we are 1 commit behind the version at GitHub:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (master)
$ git log --oneline -n 5
92ad3b7 adding "Docs/Util - Git commands used to create this version.txt"
68efb0e Minor changes for TM+SI version (different Cassini port and README.md)
6373d97 adding Library files
d766b6e Adding Android and IOS library files
89dbd47 adding v3.1 of the ReleaseNotes.v3.1.txt
```

Let's create a new branch to do the updates (just in case)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (master)
$ git checkout -b 32_Final_Update
Switched to a new branch '32_Final_Update'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$
```

And confirm that we are also not in sync with the *TeamMentor/Master* repository

PUBLIC TeamMentor / Master

Code Network Pull Requests 0 Issues 84 Wiki Graphs Admin

branch: master Files Commits Branches 4 Tags 3 Downloads 5

Master / Commit History [RSS](#)

Keyboard shortcuts available

Oct 25, 2012

- Fixing login-loop on /debug page DinisCruz authored 2 days ago 4aae621eb + Browse code ↗
- changed size of image on 'About page' (to avoid scrolling) DinisCruz authored 2 days ago 7011aa7c2f + Browse code ↗
- Added Copyright statement to both popup and right-panel article viewer ... DinisCruz authored 2 days ago 6c94cb09d0 + Browse code ↗

Oct 15, 2012

- set version to TM 3.2.2 DinisCruz authored 12 days ago cb4366da9b + Browse code ↗
- Added Fix to deal with re-authentication loop that was happening on A... DinisCruz authored 12 days ago a8b3c45164 + Browse code ↗

Here are the current remotes (in *TeamMentor_SI_Library*):

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git remote -v
origin  git@github.com:TeamMentor/TeamMentor_SI_Library.git (fetch)
origin  git@github.com:TeamMentor/TeamMentor_SI_Library.git (push)
tm_library  git@github.com:TeamMentor/Library_SI.git (fetch)
tm_library  git@github.com:TeamMentor/Library_SI.git (push)
tm_master  https://github.com/TeamMentor/Master (fetch)
tm_master  https://github.com/TeamMentor/Master (push)

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$
```

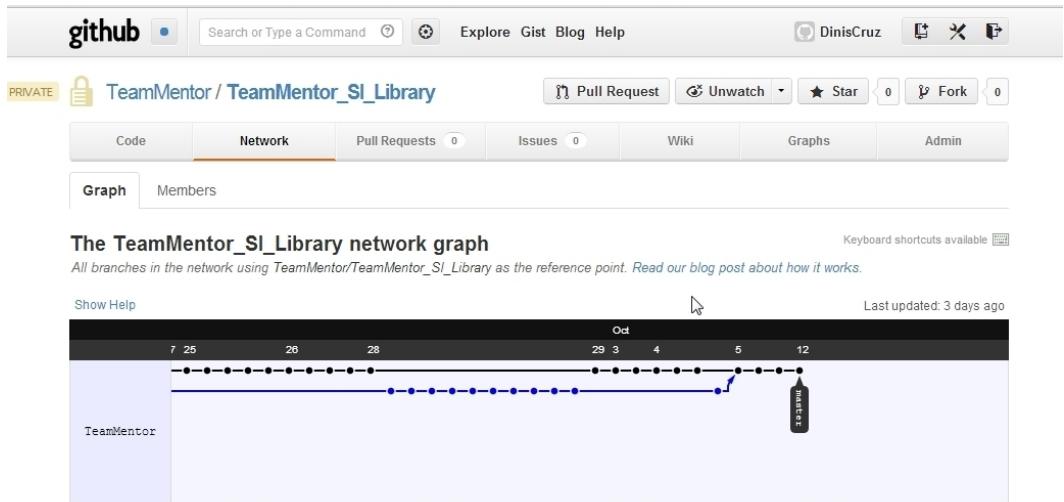
Here is gitk visualization of the *TeamMentor_SI_Library* commits

File Edit View Help

32 Final Update master remotes/origin/master adding 'Docs/Util - Git commands used to cre...
 Minor changes for TM+SI version (different Cassini port and README.md)
 adding Library files
 remotes/tm_library/master Adding Android and IOS library files
 Links Fixes on all Libraries
 Html save normalization
 2012Q2Content changes - adding Android and iOS library
 2012Q2Content changes - removing .NET 2.0 library
 2012Q2Content changes to PHP
 2012Q2Content changes to PCI DSS Compliance
 2012Q2Content changes to Java
 2012Q2Content changes to C++
 2012Q2Content changes to .NET 4.0
 2012Q2Content changes to .NET 3.5
 Removed the two lonely files.
 Removed .NET and Windows stuff from the Java library.
 Restore CWE to Serge Edition.
 Small fix to a hyperlink in one of the articles.

DinisCruz <dinis.cruz@owasp.org>
 SergeTruth <struth@securityinnovations.com>
 SergeTruth <struth@securityinnovations.com>
 SergeTruth <struth@securityinnovations.com>
 SergeTruth <struth@securityinnovations.com>

Here is GitHub visualization of the **TeamMentor_SI_Library** commits



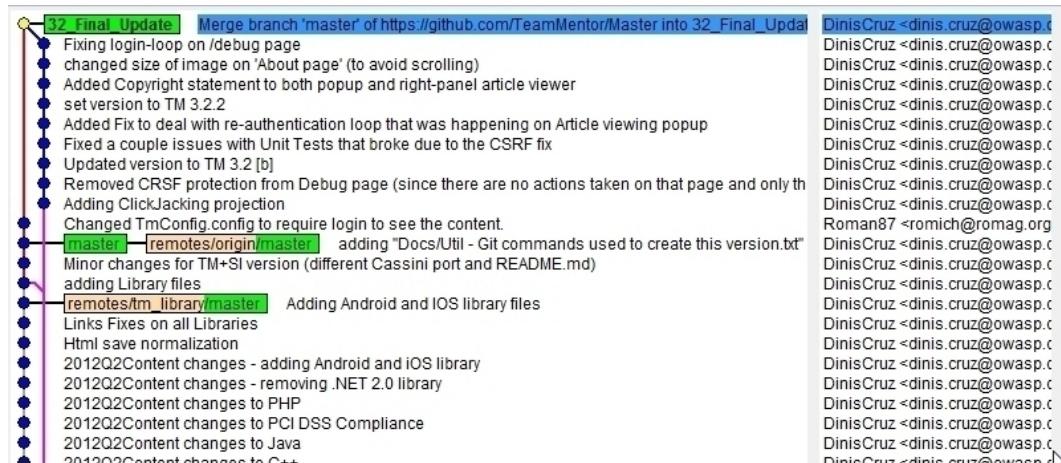
We're now ready to do the pulls (fetch+merge), and let's start with the **TeamMentor_SI_Library** (origin remote)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git pull origin master
remote: Counting objects: 9, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 5 (delta 4), reused 5 (delta 4)
Unpacking objects: 100% (5/5), done.
From github.com:TeamMentor/TeamMentor_SI_Library
 * branch           master      -> FETCH_HEAD
Updating 92ad3b7..b0eed10
Fast-forward
 Web Applications/TM_Website/TmConfig.config |    2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Next lets do the *TeamMentor/Master* (tm_master remote)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git pull tm_master master
remote: Counting objects: 166, done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 120 (delta 97), reused 69 (delta 46)
Receiving objects: 100% (120/120), 138.47 KiB, done.
Resolving deltas: 100% (97/97), completed with 40 local objects.
From https://github.com/TeamMentor/Master
 * branch           master      -> FETCH_HEAD
Removing Web Applications/packages/TeamMentor.CoreLib.3.2/TeamMentor.CoreLib.3.2.nuspec
Removing Web Applications/TeamMentor.CoreLib/TeamMentor.CoreLib.nuspec
Merge made by the 'recursive' strategy.
 Web Applications/TM_Website/Aspx_Pages/Debug.aspx |    3 +-+
 .../Html_Pages/Gui/Panels/About_TeamMentor.html |    2 +-+
 .../Html_Pages/Gui/Panels/Right_GuidanceItem.html |   17 +-+
 .../GuidanceItemViewer/GuidanceItemViewer.html |   12 +-+
 .../html/QUnit - TM_WebService_User.html |    5 +-+
 .../html/QUnit - TM_XmlDatabase.html |    5 +-+
 .../js/TM_Gui/TM_GUI_LibraryTree - Edit Mode.js |   16 +-+
 .../TM_qUnit_Tests/js/_TM_QUnit_support.js |    8 ++
 .../TM_Website/Javascript/TM/GlobalVariables.js |    8 ++
 .../TM_Website/Javascript/TM/Settings.js |    2 ++
 .../TM_Website/Properties/AssemblyInfo.cs |    33 ++
 Web Applications/TM_Website/TM_Website.csproj | 1319 ++++++-----+
 Web Applications/TM_Website/TM_Website.csproj.user |  58 ++
 .../TM_Website/bin/TeamMentor.CoreLib.dll |  Bin 354304 -> 355328 bytes
 .../TeamMentor.CoreLib/Properties/AssemblyInfo.cs |  20 ++
 .../TM_AppCode/Authentication/TM_Authentication.cs |  10 ++
 .../TeamMentor.CoreLib/TM_AppCode/Global.asax.cs |    3 ++
 .../HttpHandlers/UrlRewrite/HandleUrlRequest.cs |   15 ++
 .../TeamMentor.CoreLib/TM_AppCode/TmConfig.cs |    3 ++
 .../TeamMentor.CoreLib/TM_AppCode/UtilMethods.cs |   72 ++
 .../TM_AppCode/WebServices/TM_WebServices.asmx.cs |   20 ++
 .../TeamMentor.CoreLib/TeamMentor.Corelib.csproj |    3 -
 .../TeamMentor.CoreLib/TeamMentor.Corelib.nuspec |   26 -
 .../TeamMentor.CoreLib/packages.config |    3 ++
 .../TeamMentor.CoreLib.3.2.1.nuspec |    29 +
 .../TeamMentor.CoreLib.3.2.nuspec |  Bin 2510 -> 0 bytes
 .../TeamMentor.Website.3.2.1.2.nuspec |  556 ++++++-----+
27 files changed, 1451 insertions(+), 797 deletions(-)
create mode 100644 Web Applications/TM_Website/Properties/AssemblyInfo.cs
delete mode 100644 Web Applications/TeamMentor.CoreLib/TeamMentor.CoreLib.nuspec
create mode 100644 Web Applications/packages/TeamMentor.CoreLib.3.2/TeamMentor.CoreLib.3.2.1.nuspec
delete mode 100644 Web Applications/packages/TeamMentor.CoreLib.3.2/TeamMentor.CoreLib.3.2.nuspec
create mode 100644 Web Applications/packages/TeamMentor.Website.3.2/TeamMentor.Website.3.2.1.2.nuspec
```

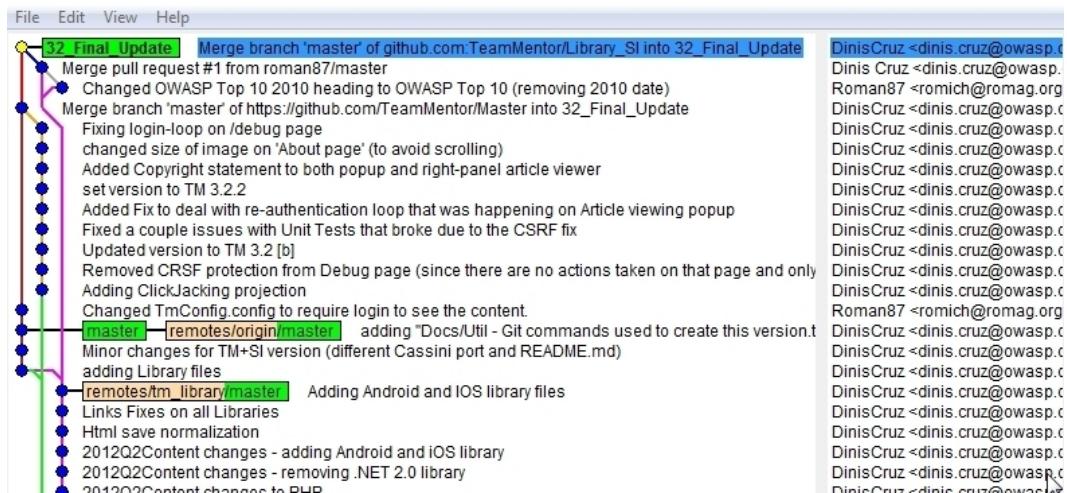
A quick look at git shows how these multiple commits are being nicely merged together



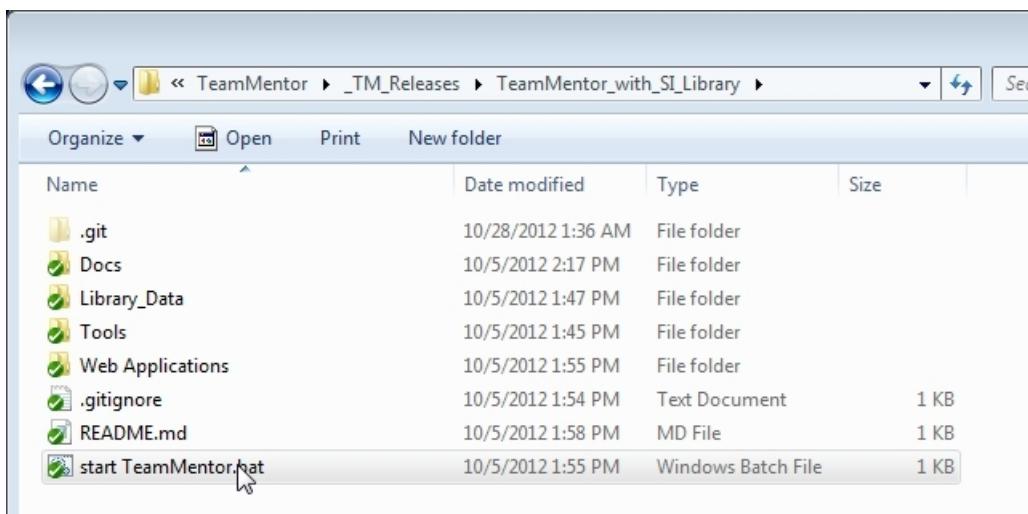
Finally lets do a git pull on the *Library_SI* (tm_library remote)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git pull tm_library master
remote: Counting objects: 20, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 11 (delta 5), reused 11 (delta 5)
Unpacking objects: 100% (11/11), done.
From github.com:TeamMentor/Library_SI
 * branch      master    -> FETCH_HEAD
Auto-merging Library_Data/XMLDatabase/TM_Libraries/PHP/PHP.xml
Auto-merging Library_Data/XMLDatabase/TM_Libraries/Java/Java.xml
Auto-merging Library_Data/XMLDatabase/TM_Libraries/.NET 4.0/.NET 4.0.xml
Auto-merging Library_Data/XMLDatabase/TM_Libraries/.NET 3.5/.NET 3.5.xml
warning: inexact rename detection was skipped due to too many files.
warning: you may want to set your merge.renamelimit variable to at least 4396 and retry the command.
Merge made by the 'recursive' strategy.
 Library_Data/XMLDatabase/TM_Libraries/.NET 3.5/.NET 3.5.xml |    2 +-
 Library_Data/XMLDatabase/TM_Libraries/.NET 4.0/.NET 4.0.xml |    2 +-
 Library_Data/XMLDatabase/TM_Libraries/Java/Java.xml |    2 +-
 Library_Data/XMLDatabase/TM_Libraries/PHP/PHP.xml |    2 +-
 4 files changed, 4 insertions(+), 4 deletions(-)
```

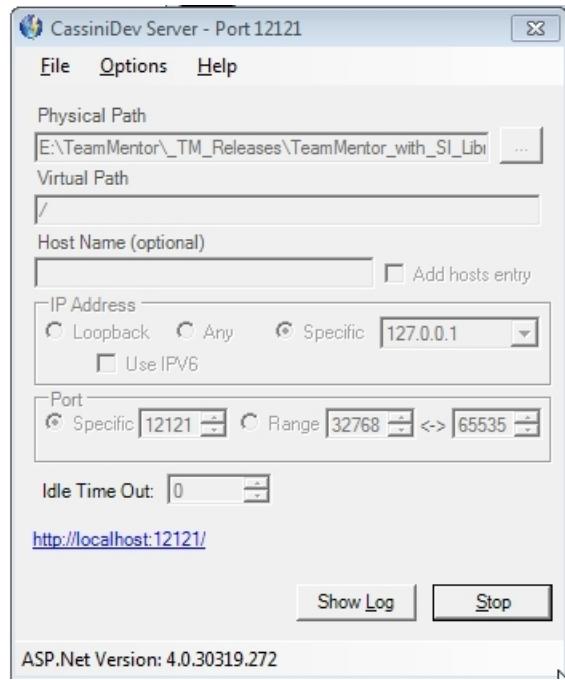
With gitk now looking like this, which is a pretty cool graph:



A look at the file system, shows that it looks as expected



So let's quickly start cassini to take a look (via the 'start TeamMentor.bat')



Hummm... on load there was a problem with the right-hand side panel

The screenshot shows the TeamMentor application running locally at 127.0.0.1:12121. The interface includes a navigation bar with links for Login, Sign Up, and About, and a version TM 3.2.3. On the left, there's a sidebar titled "Applied Filters" containing a "Guidance Libraries" tree view with categories like .NET 3.5, .NET 4.0, Android, C++, CWE, iOS, Java, PCI DSS Compliance, and PHP. The main content area has two filter panels: "Technology" (Any, .NET 3.5, ASP.NET 3.5, WCF 3.5, Web Application) and "Phase" (Deployment, Design, Implementation, Test). Below these is a message "Loaded 1780 out of 1780". A table lists three audit items: "A Centralized Log Server Is Deployed" (Title), "A Connection Is Used to the Database with a Least-privileged Service Account" (Title), and "A Control Flow Analysis Is Performed" (Title). At the bottom, a developer toolbar shows network activity and a console log with XHR requests and a failed GET request for a scriptCombiner file.

The error was this one (which is the first time I saw it)

Server Error in '/' Application.

Specified argument was out of the range of valid values.

Parameter name: utcDate

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.ArgumentOutOfRangeException: Specified argument was out of the range of valid values.
Parameter name: utcDate

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

```
[ArgumentOutOfRangeException: Specified argument was out of the range of valid values.  
Parameter name: utcDate]  
System.Web.HttpCachePolicy.UtcSetLastModified(DateTime utcDate) +3175458  
System.Web.StaticFileHandler.ProcessRequestInternal(HttpContext context, String overrideVirtualPath) +1130  
System.Web.DefaultHttpHandler.BeginProcessRequest(HttpContext context, AsyncCallback callback, Object state) +347  
System.Web.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute() +8970356  
System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously) +184
```

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.0.30319.272

Luckily a quick google search, revealed these articles:

- [http://stackoverflow.com/questions/2782329/specifyd-argument-was-out-of-the-range-of-valid-values-parameter-name-utcdate³⁷](http://stackoverflow.com/questions/2782329/specifyd-argument-was-out-of-the-range-of-valid-values-parameter-name-utcdate)
- [http://weblogs.asp.net/hajan/archive/2011/10/30/time-travel-issue-specifyd-argument-was-out-of-the-range-of-valid-values-parameter-name-utcdate.aspx³⁸](http://weblogs.asp.net/hajan/archive/2011/10/30/time-travel-issue-specifyd-argument-was-out-of-the-range-of-valid-values-parameter-name-utcdate.aspx) (screenshot below)

Which pointed me to the fact that I hit a weird ‘time bug’ that happens when the clock moves due to Daylight Saving Time.

³⁷<http://stackoverflow.com/questions/2782329/specifyd-argument-was-out-of-the-range-of-valid-values-parameter-name-utcdate>

³⁸<http://weblogs.asp.net/hajan/archive/2011/10/30/time-travel-issue-specifyd-argument-was-out-of-the-range-of-valid-values-parameter-name-utcdate.aspx>

<http://weblogs.asp.net/hajan/archive/2011/10/30/time-travel-issue-specified-argument-was-out-of-the-range-of-valid-values.aspx>

```
[ArgumentOutOfRangeException: Specified argument was out of the range of valid values.
Parameter name: utcDate]
   System.Web.HttpCachePolicy.UtcSetLastModified(DateTime utcDate) +3183846
   System.Web.StaticFileHandler.ProcessRequestInternal(HttpContext context, String overrideVirtualPath) +1130
   System.Web.DefaultHttpHandler.BeginProcessRequest(HttpContext context, AsyncCallback callback, Object state) +347
   System.Web.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute() +8963444
   System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously) +184
```

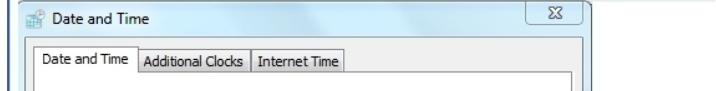
Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.0.30319.237

The very first moment I was like... ops... what the hell... however, after several seconds I just looked at my clock and saw the current time is 02:04 AM. Why is important my current time with relation to the issue??? Well, it is because five minutes ago the time was 02:59... You wonder what happened? The clock got set back 1 hour because the Daylight Saving Time ended several minutes ago, October 30, at 3:00 AM (gets back 1 hour to 2:00 AM).

Ok. That's perfectly fine, but why this confusing problem appears?

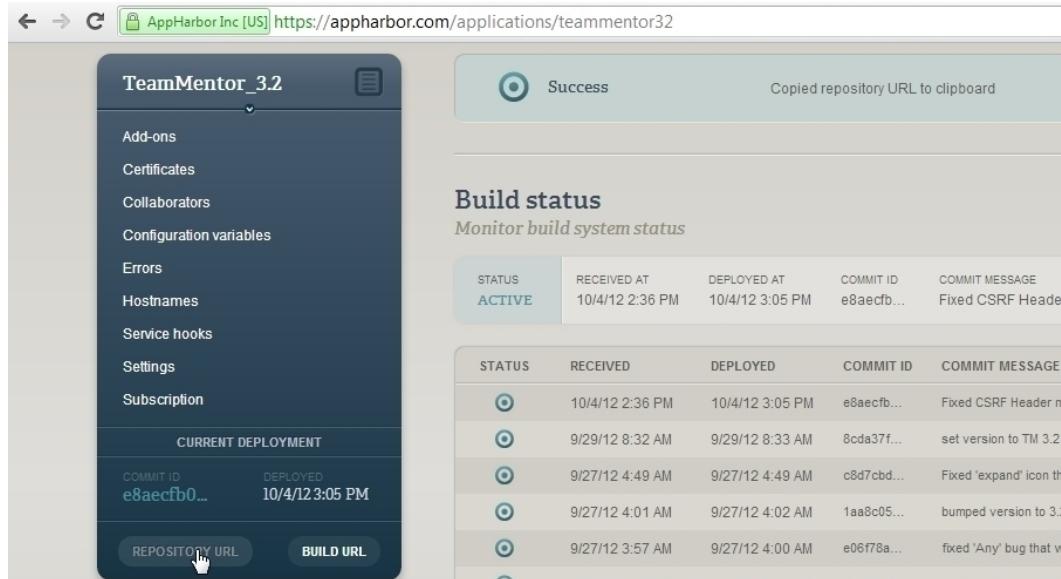
The answer is simple... The assembly was created with time 02:59 AM, 30 October, 2011. The time now is 02:04 AM, 30 October, 2011, which means we have traveled in the future :-)... hah, in other words, the assembly was last modified in future time, therefore the utcDate is out of the range of valid values.

A fast solution is to either change your time in the future or best would be to uncheck 'Automatically adjust clock for Daylight Saving Time'



Next step is to try the AppHarbor (Cloud) deployment, so I went to AppHarbor and copied the target Application **Repository URL**

<https://appharbor.com/applications/teammentor32>



Status	RECEIVED AT	DEPLOYED AT	COMMIT ID	COMMIT MESSAGE
ACTIVE	10/4/12 2:36 PM	10/4/12 3:05 PM	e8aecfb...	Fixed CSRF Header n
Success	10/4/12 2:36 PM	10/4/12 3:05 PM	e8aecfb...	Fixed CSRF Header n
Success	9/29/12 8:32 AM	9/29/12 8:33 AM	8cda37f...	set version to TM 3.2
Success	9/27/12 4:49 AM	9/27/12 4:49 AM	c8d7cbd...	Fixed 'expand' icon th
Success	9/27/12 4:01 AM	9/27/12 4:02 AM	1aa8c05...	bumped version to 3.2
Success	9/27/12 3:57 AM	9/27/12 4:00 AM	e08f78a...	fixed 'Any' bug that w
Success	9/26/12 4:16 PM	9/26/12 4:50 PM	2582640...	Fixed 'Any' bug that w

Which I added as a remote, and used to push the content

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git remote add appharbor https://DinisCruz@appharbor.com/teammentor32.git
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git push appharbor
Password for 'https://DinisCruz@appharbor.com':
To https://DinisCruz@appharbor.com/teammentor32.git
  e8aecfb..92ad3b7  master -> master
```

This built ok, but when I deployed it, I noticed that the site was still on version 3.2 (instead of the new 3.2.3)

The screenshot shows the AppHarbor build status interface. At the top, a success message says "Build '92ad3b7656c76f85916df14f21fda41ffb9e1c14' is queued for deployment". Below this, there's a "Build status" section with a "Monitor build system status" link. A table lists builds with columns: STATUS, RECEIVED AT, DEPLOYED AT, COMMIT ID, and COMMIT MESSAGE. One build is active, and another is listed as received. A "DEPLOY" button is visible next to the second build.

STATUS	RECEIVED AT	DEPLOYED AT	COMMIT ID	COMMIT MESSAGE
ACTIVE	10/4/12 2:36 PM	10/4/12 3:05 PM	e8aecfb...	Fixed CSRF Header name which was preventing u...
○	10/28/12 1:54 AM		92ad3b7...	adding "Docs/Util - Git commands use..."
○	10/4/12 2:36 PM	10/4/12 3:05 PM	e8aecfb...	Fixed CSRF Header name which was...

Back in git, I realize that I had pushed the wrong branch, and bellow I'm pushing the 32_Final_Update branch into the AppHarbor master branch (which is automatically built)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git push appharbor 32_Final_Update:master
Password for 'https://DinisCruz@appharbor.com':
To https://DinisCruz@appharbor.com/teammentor32.git
  92ad3b7..8947f49  32_Final_Update -> master
```

With the correct version ready, I deployed it:

The screenshot shows a 'Build status' section with the sub-header 'Monitor build system status'. It displays deployment details: RECEIVED AT 10/28/12 12:54 AM, DEPLOYED AT 10/28/12 12:58 AM, and COMMIT ID 92ad3b... with a magnifying glass icon. Below this is a table with columns STATUS, RECEIVED, DEPLOYED, COMMIT ID, and COMMIT MESSAGE. Two rows are shown: one for a deployment at 1:01 AM with commit 8947f49... and message 'Merge branch 'master' of github.c...', and another for a deployment at 12:54 AM with commit 92ad3b7... and message 'adding "Docs/Util - Git commands ...'. A 'DEPLOY' button is visible.

And here she is in action:

The screenshot shows the TeamMentor application interface. The top navigation bar includes links for 'Login', 'Sign Up', 'TM 3.2.3', and 'About'. On the left, there's a sidebar with 'Applied Filters' and 'Guidance Libraries'. The main content area has a search bar and three filter dropdowns: 'Technology', 'Phase', and 'Type'. Below these filters, a message says 'Showing 0 items (out of 0)'. At the bottom of the main content area, it says 'No data available in table'. To the right, a 'Selected Guidance Item' panel displays a 'Login Required' message, stating 'You must be logged in to view this page. Please [Login](#) or [Sign Up](#)'.

The reason why there is no content is because AppHarbor will only copy to the live servers the files it can find using the VisualStudio solution, which means that the Library files were not there.

To make the test realistic, I zipped the TM_Libraries folder

The screenshot shows a file explorer window with the path 'TeamMentor_with_SI_Library > Library_Data > XmlDatabase'. The toolbar includes 'Organize', 'Open', 'Include in library', 'Share with', and 'New folder'. The file list table has columns for Name, Date modified, Type, and Size. It shows two entries: 'TM_Libraries' (File folder) last modified on 10/5/2012 1:57 PM and 'TM_Libraries.zip' (ZIP File) last modified on 10/28/2012 1:04 AM, with a size of 13,103 KB.

And used TeamMentor's Control Panel to upload the file:

Install Library

Library Zip files uploaded to this server (click to install)

Current Libraries Zip Folder(server side):
D:\websites\0a\6f258\16036\0x0001_PublishedWebsites\TM_Website\App_Data\Library_Data\XmlDatabase\TM_Library_UploadedFiles

Note that instalation from Zip will override existing libraries.

Install Library from File or Url

URLs for Libraries to install: [OWASP Top 20 Vulnerabilities](#)

File (server-side) or Url:

Unzip Password:

Install

Upload Library to server

Upload a file

• TM_Libraries.zip 75% from 12.8MB [Cancel](#)

The screenshot shows the 'Install Library' section of the TeamMentor Control Panel. It includes fields for entering URLs or file paths and unzipping passwords. A large 'Install' button is present. Below this is the 'Upload Library to server' section with a red 'Upload a file' button. A progress bar at the bottom indicates the upload of 'TM_Libraries.zip' is at 75% completion from a 12.8MB file. To the right of the main interface is a file explorer window titled 'TeamMentor_with_SI_Library'. The 'Library_Data\XmlDatabase' folder is selected, showing four files: 'TM_Libraries' (File folder), 'TM_Libraries.zip' (ZIP File), 'TmUsers.xml' (XML File), and 'TmUsers_Passwords.xml' (XML File). The 'Date modified' column shows the creation times for each file.

And install it:

Install

> Library installed was successful

Upload Library to server

Upload a file

• TM_Libraries.zip 12.8MB

The screenshot shows the results of the library installation. A green success message 'Library installed was successful' is displayed above the 'Upload Library to server' section. The 'Upload a file' button is visible, and a progress bar at the bottom shows the upload of 'TM_Libraries.zip' is complete at 12.8MB.

After that, the home page looks like it should:

The screenshot shows the TeamMentor application interface. At the top, there's a navigation bar with links for 'Edit Mode', 'Control Panel', and 'Logout'. It also displays the version 'TM 3.2.3' and the user 'Logged in as: admin About'. On the left, there's a sidebar titled 'Applied Filters' under 'Guidance Libraries'. The sidebar lists several categories like '.NET 3.5', 'Fundamentals of Security', 'OWASP Top 10', etc. Below this is another sidebar for 'Technology', 'Phase', and 'Type' filters. The main content area shows a search results grid with columns for 'Title', 'Technology', and 'Phase'. The results are filtered by Technology: Any, Phase: Deployment, and Type: Implementation. The first result is 'A Centralized Log Server Is Deployed'. To the right, a large section titled 'Selected Guidance Item' displays the details for this specific item, including its title, applicable requirements (PCI DSS Requirements 10.5 through 10.7.b), what to check for (a centralized log server should be deployed), and a note that it requires login.

Testing the need to login to see the Article's content:

This screenshot shows the same TeamMentor interface as above, but with a different view. The 'Selected Guidance Item' section is now displaying a message: 'Login Required'. It states that you must be logged in to view this page and provides links for 'Login' and 'Sign Up'. The rest of the interface, including the sidebar filters and the search results grid, remains the same.

Next step is to push into the ***TMClients/TeamMentor_SI_Library*** repository this latest version

The screenshot shows the GitHub interface for creating a new repository. The repository name is set to "TeamMentor_SI_Library". The description field contains "TeamMentor with SI Library (for SI Customers)". The visibility is set to "Private". A checkbox for initializing the repository with a README is checked, and the "Add .gitignore: None" dropdown is selected. A large green "Create repository" button is at the bottom.

The screenshot shows the GitHub repository page for "TeamMentor_SI_Library". The repository details are visible, including the owner "TMClients". A "Quick setup" section provides instructions and links for "Setup in Windows" or "HTTP" and "SSH" access. The URL shown is `git@github.com:TMClients/TeamMentor_SI_Library.`

But not before we update the local ***master*** branch with the ***32_Final_Update*** branch

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (32_Final_Update)
$ git checkout master
Switched to branch 'master'
Fast-forward
.../XmlDatabase/TM_Libraries/.NET 3.5/.NET 3.5.xml | 2 ++
.../XmlDatabase/TM_Libraries/.NET 4.0/.NET 4.0.xml | 2 ++
.../XmlDatabase/TM_Libraries/Java/Java.xml | 2 ++
Library_Data/XmlDatabase/TM_Libraries/PHP/PHP.xml | 2 ++
Web Applications/TM_Website/Aspx_Pages/Debug.aspx | 3 ++
.../Html_Pages/Gui/Panels/About_TeamMentor.html | 2 ++
.../Html_Pages/Gui/Panels/Right_GuidanceItem.html | 17 ++
.../GuidanceItemViewer/GuidanceItemViewer.html | 12 ++
.../html/QUnit - TM_WebService_User.html | 5 ++
.../html/QUnit - TM_XmlDatabase.html | 5 ++
.../js/TM_Gui/TM_GUI_LibraryTree - Edit Mode.js | 8 ++
.../TM_qUnit_Tests/js/_TM_QUnit_support.js | 16 ++
.../TM_Website/Javascript/TM/GlobalVariables.js | 8 ++
.../TM_Website/Javascript/TM/Settings.js | 2 ++
.../TM_Website/Properties/AssemblyInfo.cs | 33 ++
Web Applications/TM_Website/TM_Website.csproj | 1319 ++++++-----
Web Applications/TM_Website/TM_Website.csproj.user | 58 ++
Web Applications/TM_Website/TmConfig.config | 2 ++
.../TM_Website/bin/TeamMentor.CoreLib.dll | Bin 354304 -> 355328 bytes
.../TeamMentor.CoreLib/Properties/AssemblyInfo.cs | 20 ++
.../TM_AppCode/Authentication/TM_Authentication.cs | 10 ++
.../TeamMentor.CoreLib/TM_AppCode/Global.asax.cs | 3 ++
.../HttpHandlers/UrlRewrite/HandleUrlRequest.cs | 15 ++
.../TeamMentor.CoreLib/TM_AppCode/TmConfig.cs | 3 ++
.../TeamMentor.CoreLib/TM_AppCode/UtilMethods.cs | 72 ++
.../TM_AppCode/WebServices/TM_WebServices.asmx.cs | 20 ++
.../TeamMentor.CoreLib/TeamMentor.CoreLib.csproj | 3 -
.../TeamMentor.CoreLib/TeamMentor.CoreLib.nuspec | 26 -
.../TeamMentor.CoreLib/packages.config | 3 ++
.../TeamMentor.CoreLib.3.2.1.nuspec | 29 +
.../TeamMentor.CoreLib.3.2.nuspec | Bin 2510 -> 0 bytes
.../TeamMentor.Website.3.2.1.2.nuspec | 556 ++++++-
32 files changed, 1456 insertions(+), 802 deletions(-)
create mode 100644 Web Applications/TM_Website/Properties/AssemblyInfo.cs
delete mode 100644 Web Applications/TeamMentor.CoreLib/TeamMentor.CoreLib.nuspec
create mode 100644 Web Applications/packages/TeamMentor.CoreLib.3.2/TeamMentor.CoreLib.3.2.1.nuspec
delete mode 100644 Web Applications/packages/TeamMentor.CoreLib.3.2/TeamMentor.CoreLib.3.2.nuspec
create mode 100644 Web Applications/packages/TeamMentor.Website.3.2/TeamMentor.Website.3.2.1.2.nuspec
```

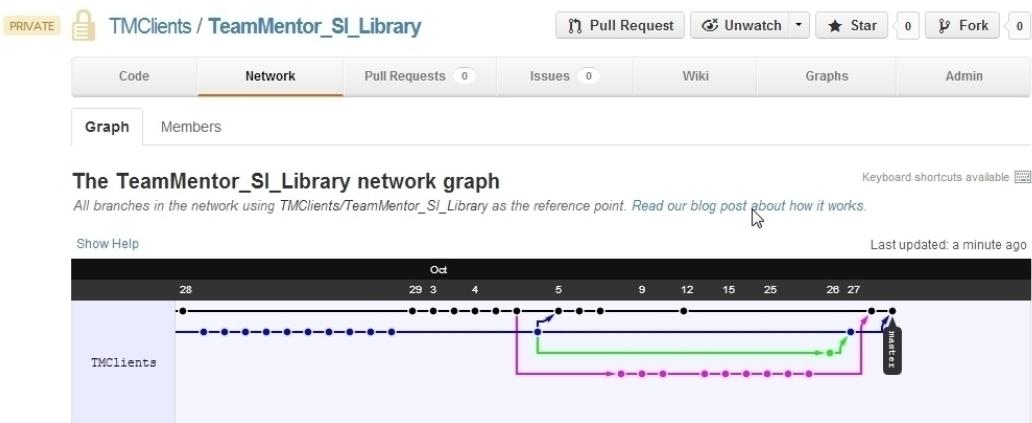
Next we replace the origin remote mapping with *TMClients/TeamMentor_SI_Library* and push into it:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (master)
$ git remote rm origin

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (master)
$ git remote add origin git@github.com:TMClients/TeamMentor_SI_Library.git

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_Releases/TeamMentor_with_SI_Library (master)
$ git push origin master
Counting objects: 30674, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (12881/12881), done.
Writing objects: 100% (30674/30674), 44.01 MiB | 582 KiB/s, done.
Total 30674 (delta 21773), reused 26569 (delta 17710)
To git@github.com:TMClients/TeamMentor_SI_Library.git
 * [new branch]      master -> master
```

After all these steps is good to take a look at the GitHub's network graph and confirm that all looks as expected:



As a final confirmation, let's download the zip file from the *TMClients/TeamMentor_SI_Library* and make sure it is all good (this is the file given to customers via a password protected zip file)



Once the file downloads, unzip it to a temp folder:

Name	Date modified	Type	Size
Docs	10/28/2012 1:21 AM	File folder	
Library_Data	10/28/2012 1:21 AM	File folder	
Tools	10/28/2012 1:21 AM	File folder	
Web Applications	10/28/2012 1:21 AM	File folder	
.gitignore	10/27/2012 6:33 PM	Text Document	1 KB
README.md	10/27/2012 6:33 PM	MD File	1 KB
start TeamMentor.bat	10/27/2012 6:33 PM	Windows Batch File	1 KB

And use 'start TeamMentor.bat' to run it locally:

The screenshot shows the TeamMentor web application interface. On the left, there's a sidebar titled 'Applied Filters' and 'Guidance Libraries'. The 'Guidance Libraries' section lists various categories such as .NET 3.5, .NET 4.0, Android, C++, CWE, IOS, Java, PCI DSS Compliance, and PHP. The main content area has three filter panels: 'Technology' (Any, .NET 3.5, ASP.NET 3.5, WCF 3.5, Web Application), 'Phase' (Deployment, Design, Implementation, Test), and 'Type' (Attack, Checklist Item, Code Example, Guideline, How To, Inspection Ques...). Below these filters, a message says 'Loaded 1780 out of 1780'. A table below shows several rows of guidance items with columns for Title, Technology, Phase, and Type. The right side of the screen has a 'Selected Guidance Item' panel with a 'Login Required' message, indicating that users must log in to view the details.

Final comment:

If you actually look at the workflow we have here, this is pretty powerful stuff!

We were able to have two complete separate activities (development and code changes) done in completely different timings, to be combined into a single package (preserving all history), that can then be delivered to customers (who don't care about the multiple repositories).

In a way it is just like doing copy and paste of the two source repositories into a ‘release folder’, but in a way that we have the full (independent) git history (check out the graphs to see how the git commits from two separate repositories are correctly preserved) and can be easily updated-synced (it took me a LOT more time to write this post than to do the actually pushes and pulls :))

What I also like about this workflow is that it works :). We are now doing the 2nd release using it, and it is surviving the real-world acid-test.

But aren’t there a lot of ‘manual’ steps that be automated? Yes, yes there are!

But I’m a big fan of:

- first figure out the solution for the problem (in a “let’s make it work” kinda way)
- then automate as much as possible the workflow that ‘works’

It’s a mistake to automate too soon, specially when the understanding of ‘the problem’ and all its moving parts is still not very high.

UPDATE: I just did the same process for the <https://github.com/TeamMentor-OWASP/Master>³⁹ version (the TeamMentor Eval), and it took less than 5m to do it :)

Remotes (of my local clone)

```

1 origin  git@github.com:TeamMentor-OWASP/Master.git (fetch)
2 origin  git@github.com:TeamMentor-OWASP/Master.git (push)
3 tm_library      git://github.com/TeamMentor-OWASP/Library_OWASP.git (fetch)
4 tm_library      git://github.com/TeamMentor-OWASP/Library_OWASP.git (push)
5 tm_master       https://github.com/TeamMentor/Master (fetch)
6 tm_master       https://github.com/TeamMentor/Master (push)

```

Commands executed:

```

1 $ git checkout -b 32_Final_Update
2 $ git pull origin master
3 $ git pull tm_library master
4 $ git pull tm_master master

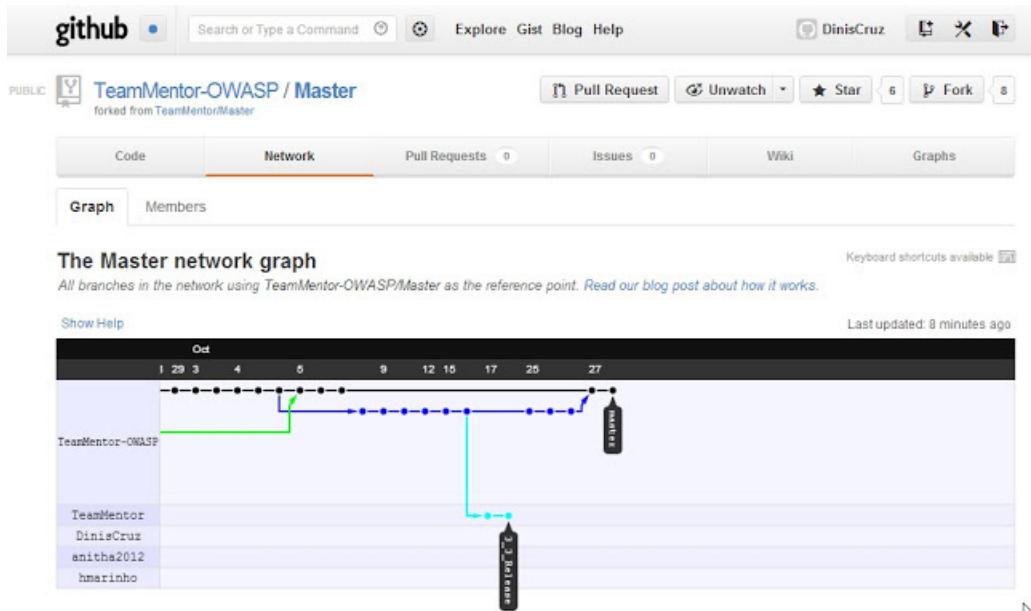
```

(there was a small conflict on the About.html page which is different on the eval version)

³⁹<https://github.com/TeamMentor-OWASP/Master>

```
1 $ git checkout master
2 $ git merge 32_Final_Update
3 $ git push origin
```

Network Graph:

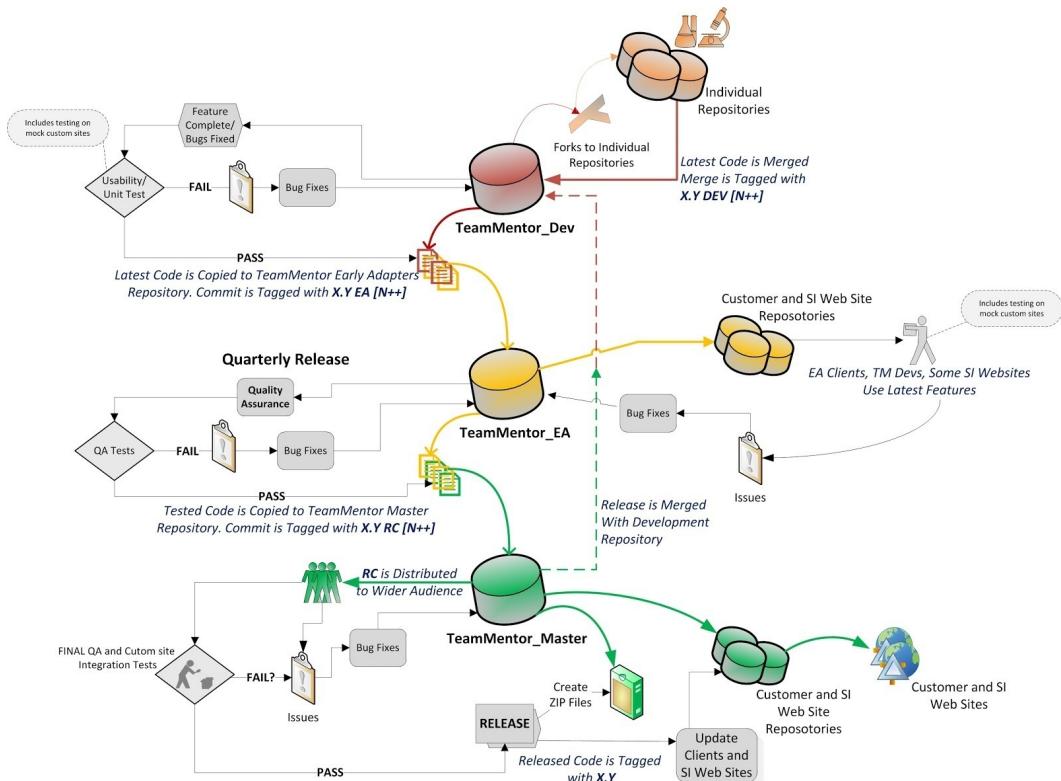


2. November 2012

- Pretty cool visualization of the ‘GitHub based’ TeamMentor Development+QA+Release workflow

Pretty cool visualization of the 'GitHub based' TeamMentor Development+QA+Release workflow

Roman created this nice Visio diagram with our current 'GitHub based' workflow for TeamMentor¹:



Although this is still a draft, it is already a good representation of how we are using Git's powerful forking/cloning capabilities to implement solid Development + QA + Release workflows.

Roman is also creating simpler versions of this diagram, for:

- the cases where we push a patch on a released version (for example targeted bugs or security fixes)

¹<http://teammentor.net/>

- the workflow used to manage the multiple TeamMentor's Content repositories

Let me know if you have any ideas or suggestions on how to make this even better :)

One area we still need to do some work (and help), is the automation of some of these steps/workflow. We have figured out the git commands, and now we need to automate their execution.

3. December 2012

- Comparing two GitHub Issues List
- Using TeamCity to build on Git Commit, deploy to AppHarbor and open browser
- Minimum required files to run git.exe on windows (for clone, push and pull)
- Rewriting Git History (locally and at GitHub)

Comparing two GitHub Issues List

Is there a way to compare two GitHub Issues lists?

What I need is a programmatic way to compare the items that exist in two GitHub repositories (repo A and repo B), do a diff, and list the ones that don't exist in repo B (i.e. which ones were not copied from repo A to repo B)

Recently we moved the TeamMentor issues/bug list into the public repository: <https://github.com/TeamMentor/TeamMentor>

Before (when started to use GitHub for TeamMentor development) we used the private repository <https://github.com/DinisCruz/TeamMentor-v3.0/issues>²

¹<https://github.com/TeamMentor/Master/issues>

²<https://github.com/DinisCruz/TeamMentor-v3.0/issues>

When we did the switch, we manually moved a number of issues into the new public repository, but I want to make sure we don't lose anything (since these issue's list are also our brain-dump of ideas for future releases)

So, any tools or services that currently do this?

Using TeamCity to build on Git Commit, deploy to AppHarbor and open browser

I just gave [TeamCity³](#) a test drive and I really LIKED it :)



Continuous Integration for Everybody

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily — leading to multiple integrations per day. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.

Martin Fowler

After using it for a bit, I was able to create a really nice [CI⁴](#) that:

- Monitors the local file system for a Git Commit (of TeamMentor)
- On Git Commit, trigger a build (of the main VisualStudio 2010 project)
- If the build is OK, trigger a git push into AppHarbor
- Open a WebBrowser with the AppHarbor site

All automated from the moment there is a commit :)

Let's see this in action

It all starts with a deployed instance of TeamMentor at version Dev.2.1 (see top right)

³<http://www.jetbrains.com/teamcity/>

⁴http://en.wikipedia.org/wiki/Continuous_integration

The screenshot shows the TeamMentor web application interface. At the top, there's a header with the TeamMentor logo, a search bar, and navigation links for 'Login', 'Sign Up', 'TM 3.3 Dev.2.1', and 'About'. On the left, there's a sidebar with 'Applied Filters' and 'Guidance Libraries'. The main area has two search fields for 'Technology' and 'Phase', both currently empty. Below them, a message says 'Showing 0 items (out of 0)'. A table header row is visible with columns for 'Title', 'Technology', 'Phase', 'Type', and 'Category'. A note at the bottom of the table area says 'No data available in table'. On the right, a panel titled 'Selected Guidance Item' displays the message 'No article available.'

With TeamCity running locally on my box (as a service) keeping an eye on the local git repository

The screenshot shows the TeamCity interface. At the top, there are tabs for 'Projects', 'My Changes', 'Agents', 'Build Queue', and user information 'Dinis Cruz | Administration'. Below the tabs, there's a tree view showing 'TeamMentor 3.3' and 'TeamMentor Local Git'. Under 'TeamMentor Local Git', it shows a build configuration '#12' with status 'Success', 'No artifacts', and 'No changes', updated '27 minutes ago (4s)'. There are also buttons for hiding successful configurations and configuration details.

In VisualStudio, lets make a simple change (bumping the version to Dev.2.2)

The screenshot shows a Visual Studio code editor window with a tab for 'Settings.js'. The code contains several lines of JavaScript, including setting 'TM.tmVersion' to 'TM 3.3 Dev.2.2', defining 'TM.ArticleTitle' as 'TeamMentor 3.3', and defining 'TM.tmWebServices' and 'TM.NotAuthorizedPage' paths. There are also comments for 'TM.Gui.GaugesID' and 'TM.Gui.showLibraryStructureToAnonymous'.

```

//TM Settings
TM.tmVersion = "TM 3.3 Dev.2.2";
TM.ArticleTitle = "TeamMentor 3.3";

TM.tmWebServices      = '/Aspx_Pages/TM_WebServices.asmx/';
TM.NotAuthorizedPage = '/Html_Pages/Gui/Panels/AD_Non_Authorized_User.html';

//TM.Gui.GaugesID
TM.Gui.showLibraryStructureToAnonymous           = null;      // set to GaugesID to enable it
TM.Gui.LoadLibraryData                          = true;
TM.Gui.editMode                                = false.

```

Using *Git Source Control Provider* VisualStudio Extension, create a Commit

Git Pending Changes - master

Commit Amend Last Commit Switch Branch Open .gitignore Refresh Git Bash Git History TortoiseGit About

Comments: Sign off

Updating version to Dev.2.2

Diff (double click to open file):

```

1 diff --git a/Web Applications/TM_Website/Javascript/
2 index dbc3889..2bcdf3 100644
3 --- a/Web Applications/TM_Website/Javascript/TM/Setting.js
4 +++ b/Web Applications/TM_Website/Javascript/TM/Setting.js
5 @@ -1,5 +1,5 @@
6 //TM Settings
7 -TM.tmVersion = "TM 3.3 Dev.2.1";
8 +TM.tmVersion = "TM 3.3 Dev.2.2";
9 TM.ArticleTitle = "TeamMentor 3.3";
10 TM.tmWebServices = '/Aspx_Pages/TM_WebServices.
11
12

```

Changed files: (master) +47 ~8 -2 !0

Status File Modified Web Applications/TM_Website/Javascript/TM/Setting.js

Which (after about 10 to 20 secs) is picked up by TeamCity

localhost:89/project.html?projectId=project2

Projects My Changes Agents Build Queue Dinis Cruz Administration

TeamMentor 3.3 (TeamMentor 3.3)

Overview Change Log Statistics Current Problems Investigations Muted Tests

TeamMentor Local Git

#	Status	Artifacts	Configurations	Time Left
#13	Running	No artifacts	dinis.cruz (1)	5s left
#12	Success	No artifacts	No changes	34 minutes ago (4s)

The 'build logs' update in real-time, and provide a lot of good info:

```
localhost:89/viewLog.html?buildId=14&buildTypeId=bt3&tab=buildLog

[07:58:01] [Web Applications\TM_Website\TM_Website.csproj] CoreCompile
[07:58:01]   [CoreCompile] Csc
[07:58:01]     [Csc] C:\Windows\Microsoft.NET\Framework\v4.0.30319\Csc.exe /noconfig /nowarn:1701,1
[07:58:02] [Web Applications\TM_Website\TM_Website.csproj] _CopyFilesMarkedCopyLocal
[07:58:02]   [_CopyFilesMarkedCopyLocal] Copy
[07:58:02]     [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]       [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]         [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]           [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]             [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]               [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                 [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                   [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                     [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                       [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                         [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                           [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                             [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                               [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                                 [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                                   [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                                     [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                                       [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                                         [Copy] Copying file from "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Application"
[07:58:02]                                         [GetCopyToOutputDirectoryItems]
[07:58:02] [Web Applications\TM_Website\TM_Website.csproj] CopyFilesToOutputDirectory
[07:58:02] [Web Applications\TM_Website\TM_Website.csproj] CopyFilesToOutputDirectory
[07:58:02]   [CopyFilesToOutputDirectory] Copy
[07:58:02]     [Copy] Copying file from "obj\Release\SecurityInnovation.TeamMentor.Website.dll" to "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\bin"
[07:58:02]   [CopyFilesToOutputDirectory] TM_Website -> C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\bin
[07:58:02]   [CopyFilesToOutputDirectory] Copy
[07:58:02]     [Copy] Copying file from "obj\Release\SecurityInnovation.TeamMentor.Website.pdb" to "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\bin"
[07:58:02] [Web Applications\TM_Website\TM_Website.csproj] PostBuildEvent
[07:58:02]   [PostBuildEvent] Exec
[07:58:02]     [Exec] rem del "C:\TeamCity\buildAgent\work\c92bb8fcf3dcaca3\Web Applications\TM_Website\bin\*.dll"
[07:58:02]     [Step 1/3] Process exited with code 0
[07:58:02] Step 2/3: Publish to appHarbour (Command Line) (running for 12s)
[07:58:02]   [Step 2/3] Starting: "C:\Program Files (x86)\Git\bin\git.exe" push appharbor master
[07:58:02]     [Step 2/3] in directory: E:\TeamMentor\_TM_Releases\TeamMentor_3_3_Dev
```

After the build completes, and the AppHarbor push is done, the target website will be opened: (notice that it is now on version Dev.2.2)

The screenshot shows a web browser window for 'teammentor-3-3.apphb.com/teamMentor'. The header includes a back/forward button, a refresh icon, and a search bar. On the right, there are links for 'Login' and 'Sign Up', and a note 'TM 3.3 Dev.2.2'. The main content area features a logo for 'TeamMentor' (a stylized 'T' inside a green circle) followed by the text 'TeamMentor' and 'a Security Innovation eKnowledge Product'. Below this is a sidebar with 'Applied Filters' and 'Guidance Libraries' sections. The central part of the page has a 'Search' field and two dropdown menus: 'Technology' and 'Phase'. A message 'Showing 0 items (out of 0)' is displayed below these. To the right, a large box titled 'Selected Guidance Item' contains the message 'No article available.' At the bottom, a footer notes 'TeamMentor @ 2007-2012 all rights reserved | a [Security Innovation](#) eKnowledge Product | [Contact Us](#)'.

For reference here are the Build Steps in TeamCity configuration page:

localhost:89/admin/editBuildRunners.html?id=buildType:bt3

Dinis Cruz | Administration

Administration › TeamMentor 3.3 Project › TeamMentor Local Git Configuration | Run ... | Build Configuration Home

Build step enabled.

Build Steps

Build Step	Description	edit	more
Visual Studio (sln)	Build file path: Web Applications/TM_Website.sln Targets: Rebuild Configuration: Release Platform: <default> Execute: Only if all previous steps were successful	edit	more
Publish to appHarbour	Command Line Command: C:\Program Files (x86)\Git\bin\git.exe push appharbor master Execute: Only if all previous steps were successful	edit	more
Open Browser	Command Line Custom script: explorer "http://teammentor-3-3.apphb.co... (and 1 more line) Execute: Only if all previous steps were successful	edit	more

[+ Add build step](#) [Reorder build steps](#)

Configuration Steps

- 1 General Settings
- 2 Version Control Settings
- 3 Build Steps (3)
- 4 Build Failure Conditions
- 5 Build Triggers (1)
- 6 Dependencies
- 7 Build Parameters
- 8 Agent Requirements

Minimum required files to run git.exe on windows (for clone, push and pull)

I want to add native Git support to TeamMentor (and O2) and don't want to ship the entire git folder structure that is installed with [msysgit](#)⁵

I found the [minimal required files to just use git-clone](#)⁶ post that implies that the only files that are needed for a git clone are :

- git-clone.exe
- git-fetch-pack.exe
- git-index-pack.exe
- git.exe
- libcurl-4.dll
- libiconv2.dll

Is this correct?

What about for git pushes, pulls and commits?

I'm sure I saw this on a tool I used the other day (which had git support), but I can't remember which one

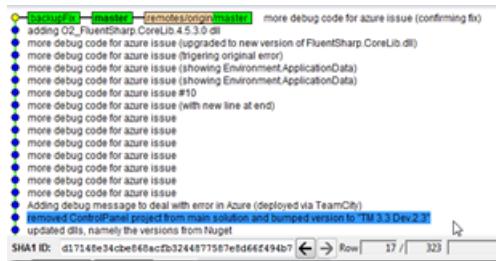
⁵<http://code.google.com/p/msysgit/>

⁶<https://groups.google.com/forum/?fromgroups#!topic/msysgit/BugV7r6eFyY>

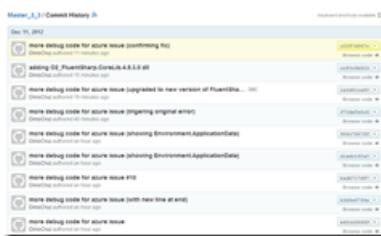
Rewriting Git History (locally and at GitHub)

When fixing the ASP.NET WCF REST help page ‘Memory gates checking’ error at AppHarbor⁷,

I ended up with a number of Git Commits: locally



and at GitHub



Once we found the solution (and pushed a new version of FluentSharp.CoreLib.dll to Nuget), it was time to clean up the history (since those commits don't really need to be in the main TM master⁸). Yes I could had used a branch, but since this was part of the TeamCity deployment tests, It was useful to do it on the master branch (and see how fast TeamCity can be :))

So what we want to do, is to **something that is not very common in Git: rewrite Git's history** (i.e. remove pushed commits). In practice this means that we want to ‘go back’ to the commit marked in blue (gitk image above), and remove the extra commits from the main Git History:

Just in case something goes wrong, lets let's backup the current changes as a local branch :)

⁷<http://blog.diniscruz.com/2012/12/problem-with-environmentspecialfolderap.html>

⁸https://github.com/TeamMentor/Master_3_3

```
$ git checkout -b backupfix
switched to a new branch 'backupfix'
$ git commit -am "Removed ControlPanel project from main solution and bumped version to 'TM 3.3 Dev 2.3'." --allow-empty
[TeamMentor_3.3_Dev: 1 commits]
$ git checkout master
switched to branch 'master'
```

Once that is done, we do a ‘forced git reset’:



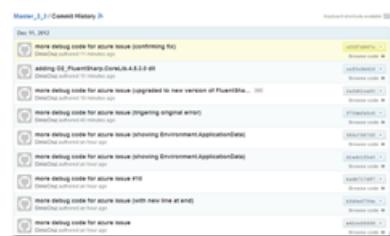
This does the trick, and now the (local) history looks good:



and



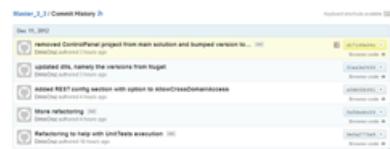
Next, to remove these commits from GitHub ...



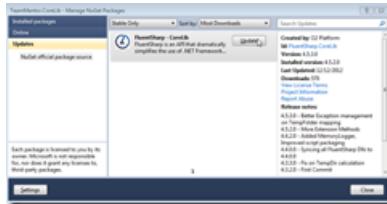
....we do a `_git push -force _`

```
$ git push --force origin master
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/TeamMentor_3.3.git
   c00f709...d1724de master -> master (Forced update)
```

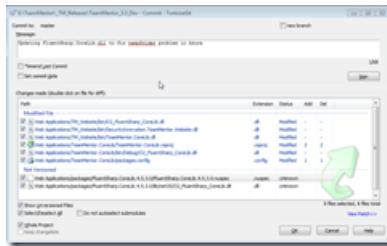
And GitHub’s version has been moved back:



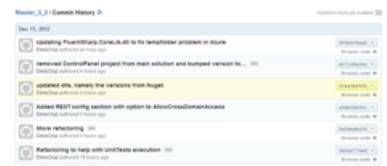
Finally we update (on the TeamMentor VS Projects) the FluentSharp.CoreLib.dll via NuGet:



and commit it:



Creating the desired Git Commit History:

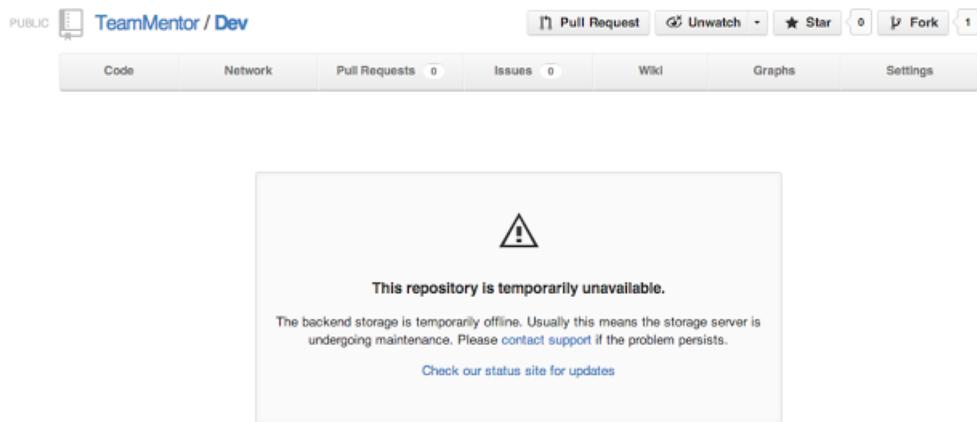


4. January 2013

- GitHub is having some probs today
- Dangerous bug between Git, GitHub and Windows (duplicate directories with different capitalization)
- Can Git be used instead of Word's 'Track Changes'

GitHub is having some probs today

Here is what a GitHub Commit page looks like:



Here is the [status page¹](#)

¹<https://status.github.com/>

github:status

UPDATED ABOUT A MINUTE AGO

10:23 UTC Some fileserver issues are recurring, we are investigating. [More Status History »](#)

What is this site?

We continuously monitor the status of [github.com](#) and all its related services. If there are any interruptions in service, a note will be posted here. If you are experiencing problems connecting to GitHub and do not see a notice posted, please email support@github.com.

GitHub.com Availability	Normal
Storage Availability	Normal
GitHub Pages Availability	Normal
API Availability	Normal
Code Downloads	Normal

And the [Status messages²](#) page:

²<https://status.github.com/messages>

github:status

UPDATED LESS THAN A MINUTE AGO

Status Messages

[« Dashboard](#)

Today

10:23 UTC Some fileserver issues are recurring, we are investigating.

10:11 UTC Everything operating normally.

9:55 UTC Were investigating issues with one of our fileservers.

Yesterday, January 09, 2013

20:03 UTC Everything operating normally.

18:11 UTC Service hooks are behind. Were working to add capacity to get these back to realtime.

1:26 UTC Everything operating normally

1:26 UTC Everything operating normally

0:05 UTC Service hooks are behind while we process through a queue backlog.

January 08, 2013

23:49 UTC Everything operating normally

23:33 UTC Services are recovering after a large database load spike.

23:29 UTC Minor service outage

23:20 UTC Major service outage

Dangerous bug between Git, GitHub and Windows (duplicate directories with different capitalization)

After doing [this rename³](#), here is what GitHub looks like:

File	Date	Commit Message
Docs	3 months ago	adding v3.1 of the ReleaseNotes.v3.1.txt [DinisCruz]
Tools	4 months ago	adding pinned functionality to Applied filters [DinisCruz]
Web Applications	2 hours ago	renamed TeamMentor.Automation project (and folder) to TeamMentor.Unit... [DinisCruz]
web applications	2 hours ago	renamed TeamMentor.Automation project (and folder) to TeamMentor.Unit... [DinisCruz]
.gitignore	3 days ago	adding bin to .gitignore [DinisCruz]
README.md	3 months ago	added ReleaseNotes for v3.2 and moved to /Docs folder [DinisCruz]
start TeamMentor.bat	3 months ago	Set version to 3.2 [DinisCruz]

Note the two Web Applications folders!!!

I have hit this issue in the past, and I think it is something to do with Windows capitalization inconsistencies (as seen on the windows search), Git and GitHub

Note how my local folder looks good.

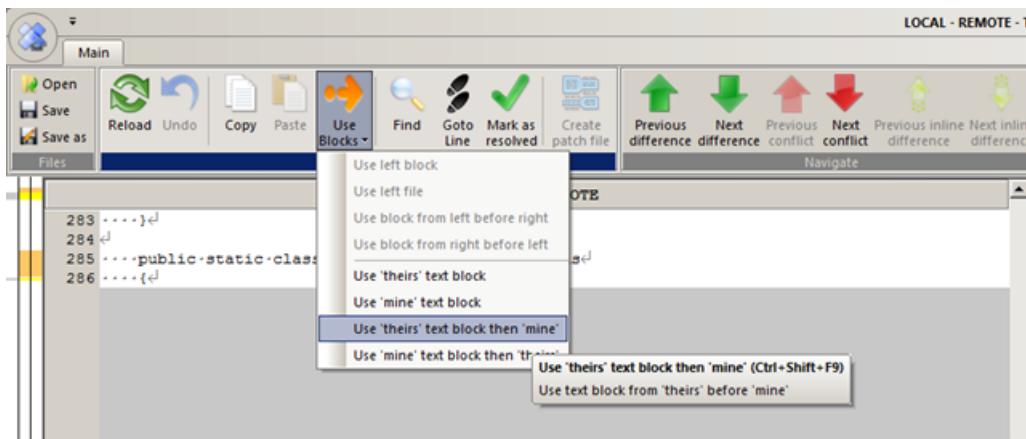
Name	Date modified	Type	Size
.git	1/10/2013 3:15 PM	File folder	
Docs	1/8/2013 1:53 AM	File folder	
Library_Data	1/10/2013 4:00 PM	File folder	
Tools	1/8/2013 1:53 AM	File folder	
Web Applications	1/10/2013 3:12 PM	File folder	
.gitignore	1/8/2013 3:00 AM	Text Document	1 KB
README.md	1/8/2013 1:53 AM	MD File	1 KB
start TeamMentor.bat	1/8/2013 1:54 AM	Windows Batch File	1 KB

Even weirder is the fact that the commit message looks good (note the renames):

³<https://github.com/michaelhidalgo/Dev/commit/2bb8505ef95969fac0a9057baf963b8c4f8e6708>



Let's try to do a git clone to see if what happens locally:

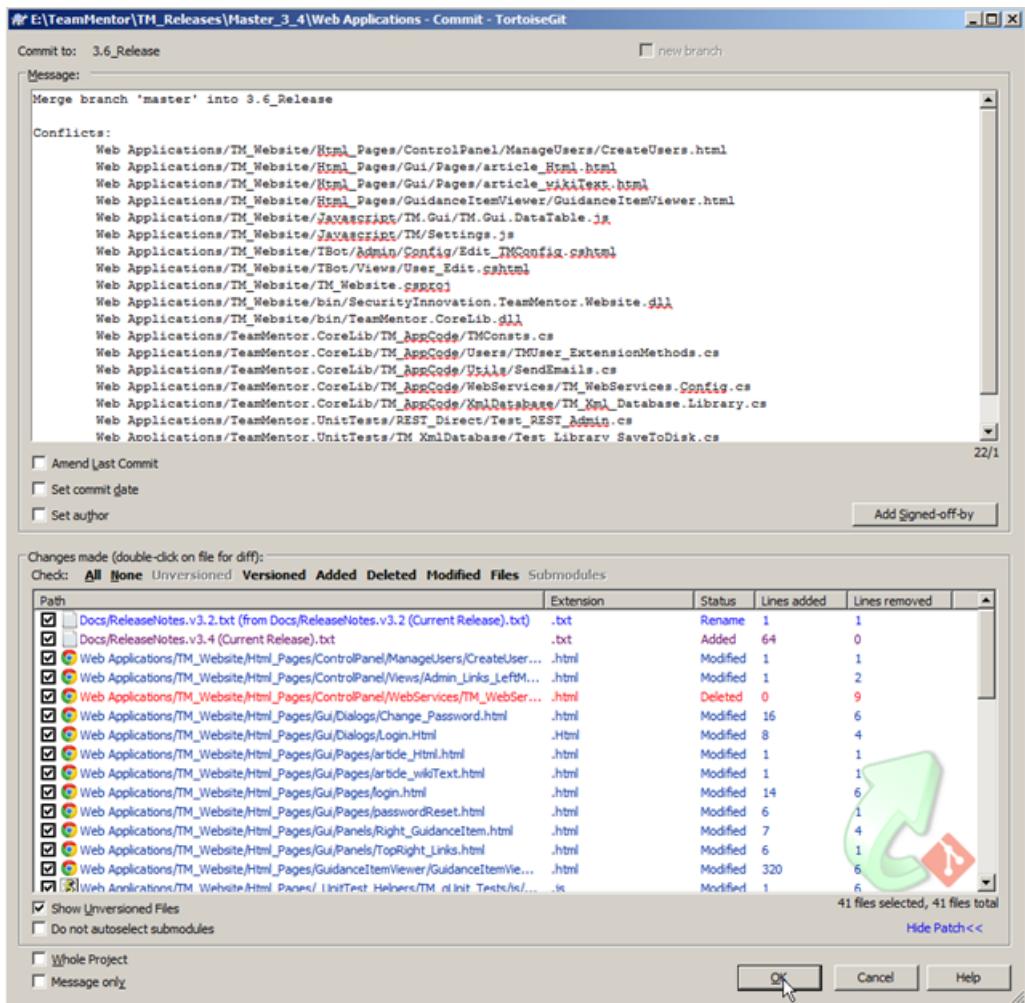


```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michael  
$ git branch -a  
* 3.3.3_Release  
  Issue_565  
  master  
  remotes/origin/Checkmarx  
  remotes/origin/HEAD -> origin/master  
  remotes/origin/HotFix_3_3_1  
  remotes/origin/Issue_142  
  remotes/origin/Issue_193  
  remotes/origin/Issue_285  
  remotes/origin/Issue_384  
  remotes/origin/Issue_400  
  remotes/origin/Issue_445  
  remotes/origin/Issue_455  
  remotes/origin/Issue_459  
  remotes/origin/Issue_461  
  remotes/origin/Issue_462  
  remotes/origin/Issue_475  
  remotes/origin/Issue_476  
  remotes/origin/Issue_481
```

Personal Access Tokens [Create new token](#)

Personal Access Tokens function like ordinary OAuth access tokens. They can be used instead of a password for git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

60396b54a6fb112eb0cc6ffbdः7f936c2b34bcc  test "" > token – Edit Delete 
iOctocat - GitHub client for iOS – Edit Delete



Humm, the clone looks good, so I wonder if this will self correct after the next commits:

```

$ git checkout 3.5_Release
Branch 3.5_Release set up to track remote branch 3.5_Release from dev.
Switched to a new branch '3.5_Release'
n20WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <3.5_Release>
$ git merge master
Updating 48c0ec0..72ca4b5
Fast-forward
  ...<Current Release>.txt => ReleaseNotes.v3.2.txt      |  2 ++
  Docs/ReleaseNotes.v3.4 <Current Release>.txt          | 64 ++++++=====
  .../ControlPanel/ManageUsers/CreateUsers.html           |  2 ++
  .../Html_Pages/Gui/Pages/article_Html.html            |  2 ++
  .../Html_Pages/Gui/Pages/article_wikiText.html        |  2 ++
  .../TM_Website/Html_Pages/Gui/Pages/login.html        | 20 +++-
  .../Html_Pages/Gui/Pages/passwordReset.html           |  7 ++
  .../Html_Pages/Gui/Panels/Right_GuidanceItem.html     |  8 ++
  .../GuidanceItemViewer/GuidanceItemViewer.html        |  2 ++
  .../Javascript/TM.Gui/TM.Gui.DataTable.js             |  2 ++
  .../TM_Website/Javascript/TM/Settings.js              |  4 ++
  .../TM_Website/Javascript/gAnalytics/ga.js           | 24 +**--_
  .../IBot/Admin/Config/Edit_IMConfig.cshtml           | 84 ++++++=====
  .../TM_Website/TBot/Views/User_Edit.cshtml           | 14 +**-
  Web Applications/TM_Website/TM_Website.csproj          |  3 ++
  Web Applications/TM_Website/Web.config                |  2 ++
  .../bin/SecurityInnovation.TeamMentor.Website.dll    | Bin 4608 -> 4608 bytes
  .../TM_Website/bin/TeamMentor.CoreLib.dll             | Bin 346751 -> 348290 bytes
  .../TM_AppCode/Authentication/TM_Authentication.cs   |  5 ++
  .../TM_AppCode/Schemas/Models/TM_User.cs              |  2 ++
  .../TeamMentor.CoreLib/TM_AppCode/TMConsts.cs         | 22 +****-
  .../TM_AppCode/Users/TMUser_ExtensionMethods.cs       | 23 +*****+
  .../TM_AppCode/Utils/SendEmails.cs                   | 16 +**-
  .../WebServices/TM_Wbservices.Config.cs               |  8 ++
  .../WebServices/TM_Wbservices.Session.cs              |  2 ++
  .../XmlDatabase/TM_Xml_Database.Library.cs           |  7 ++
  .../Asmx_WebServices/Test_Libraries.cs                | 20 +**-
  .../Test_VIEWS_Folders_Articles.cs                  | 25 +*****+
  .../CoreLib/Test_HandleRequest_LibraryData.cs        |  4 ++
  .../TeamMentor.UnitTesting/CoreLib/Test_Schemas.cs    |  4 ++
  .../TeamMentor.UnitTesting/Messaging/Test_Email.cs    | 32 +*****+
  .../REST_Direct/Test_REST_Admin.cs                  |  3 ++
  .../TM_XmlDatabase/Test_Library_SaveToDisk.cs        |  2 ++
  Web Applications/build.bat                          |  1 +
34 files changed, 345 insertions(+), 73 deletions(-)
rename Docs/ReleaseNotes.v3.2 <Current Release>.txt => ReleaseNotes.v3.2.txt (94%)
create mode 100644 Docs/ReleaseNotes.v3.4 <Current Release>.txt
create mode 100644 Web Applications/TM_Website/IBot/Admin/Config/Edit_IMConfig.cshtml
create mode 100644 Web Applications/build.bat

```

Question: Any ideas on how to fix this?

My only thought at the moment is to:

- a) delete that folder (from Git)
- b) commit and push the delete
- c) add that folder
- d) commit and push

Can Git be used instead of Word's 'Track Changes'

Absolutely.

Text changes are just a simplified version of source code :)

Here are a number of really amazing ‘non-code’ stuff that is happening with Git’s content-versioning capabilities:

- Bundes-Git – German Laws on GitHub⁴
- GitLaw: GitHub for Laws and Legal Documents - a Tourniquet for American Liberty⁵ (with Y’s discussion here⁶)
- Using Git for writing thesis [closed]⁷
- Could Git be used to track bills in Congress?⁸
- A must watch TED talk about GIT and democracy⁹

⁴<http://okfnlabs.org/blog/2012/12/13/bundesgit-german-laws-on-github.html>

⁵<http://blog.abevaelker.com/gitlaw-github-for-laws-and-legal-documents-a-tourniquet-for-american-liberty/>

⁶<http://news.ycombinator.com/item?id=3967921>

⁷<http://stackoverflow.com/questions/7775881/using-git-for-writing-thesis>

⁸<http://www.quora.com/Could-Git-be-used-to-track-bills-in-Congress>

⁹<http://blog.diniscruz.com/2012/10/a-must-watch-ted-talk-about-git-and.html>

5. March 2013

- Seeing an NGit Diff by using reflection to access the internal Sharpen.ByteArrayOutputStream Class
- GitHub.com needs to improve their the ‘Normal’ status definition and error reporting dashboard
- Another GitHub ‘Normal’ status that doesn’t allow me to Push
- Prob with (older version of) NGit where it was failing to create Git repositories in Azure/TeamCity
- Creating a version TeamMentor which uses the new GitUserData.config file
- Changing the ‘View TM article by anonymous users’ status via GitHub
- Creating a new TeamMentor test site using TeamCity, GitHub and Azure
- Using Git Branches to fix Issues added to TeamMentor’s GitHub repository
- Extracting content files from a Azure deployed version of TeamMentor (pre 3.3 git support), starting with a failed SFTP attemp and ending with a CSharp REPL script
- Using NGit to create native Git support in Azure deployed app (with automatic pushes and pulls)

Seeing an NGit Diff by using reflection to access the internal Sharpen.ByteArrayOutputStream Class

I was trying to get the NGit diff output stream, but hit on an issue that the *Sharpen.ByteArrayOutputStream* class is internal

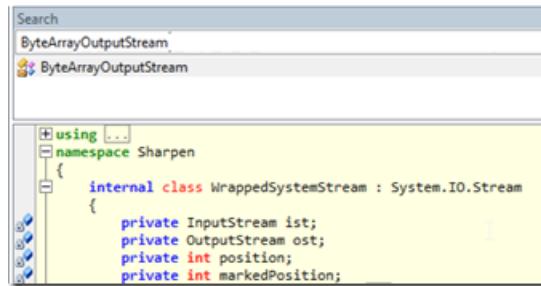
Here is an example from NGit UnitTests on how to use the NGI Diff Command:

```
[NUnit.Framework.Test]
public virtual void TestDiffModified()
{
    Write(new FilePath(db.WorkTree, "test.txt"), "test");
    FilePath folder = new FilePath(db.WorkTree, "folder");
    folder.Mkdir();
    Write(new FilePath(folder, "folder.txt"), "folder");
    Git git = new Git(db);
    git.Add().AddFilepattern(".").Call();
    git.Commit().SetMessage("Initial commit").Call();
    Write(new FilePath(folder, "folder.txt"), "folder change");
    OutputStream @out = new ByteArrayOutputStream();
    IList<DiffEntry> entries = git.Diff().SetOutputStream(@out).Call();
    NUnit.Framework.Assert.AreEqual(1, entries.Count);
    NUnit.Framework.Assert.AreEqual(DiffEntry.ChangeType.MODIFY, entries[0].GetChangeType
        ());
    NUnit.Framework.Assert.AreEqual("folder/folder.txt", entries[0].GetOldPath());
    NUnit.Framework.Assert.AreEqual("folder/folder.txt", entries[0].GetNewPath());
    string actual = @out.ToString();
    string expected = "diff --git a/folder/folder.txt b/folder/folder.txt\n" + "index 0119635..95c4c65 100644\n"
        + "--- a/folder/folder.txt\n" + "+++ b/folder/folder.txt\n" + "@@ -1 +1 @@\n" +
        "-folder\n" + "\\ No newline at end of file\n" + "+folder change\n" + "\\ No newline at end of file\n";
    NUnit.Framework.Assert.AreEqual(expected.ToString(), actual);
}
```

The key part is:

```
OutputStream @out = new ByteArrayOutputStream();
IList<DiffEntry> entries = git.Diff().SetOutputStream(@out).Call();
NUnit.Framework.Assert.AreEqual(1, entries.Count);
NUnit.Framework.Assert.AreEqual(DiffEntry.ChangeType.MODIFY, entries[0].GetChangeType
    ());
NUnit.Framework.Assert.AreEqual("folder/folder.txt", entries[0].GetOldPath());
NUnit.Framework.Assert.AreEqual("folder/folder.txt", entries[0].GetNewPath());
string actual = @out.ToString();
```

Note how the *Sharpen.ByteArrayOutputStream* was created and used on ** *Diff.SetOutputStream*, but (as we will see below), we will have a problem because this class is ***internal***:



On an O2 Platform¹ C# REPL script², lets create a quick repo and a valid Diff result:

```

run new open save as search:
Command To Execute
1 var tempRepo = "tempRepo".tempDir();
2 var nGit = tempRepo.git_Init();
3
4 var file = "testFile.txt";
5 nGit.writeFile(file, "some content\naaaa\n");
6 nGit.add(".", false).commit_using_Status();
7 nGit.writeFile(file, "some content\nna Change\n");
8
9 var diff = nGit.Git.Diff();
10 return diff.Call();

```

Invoke and Result

Execute

stop execution

Output

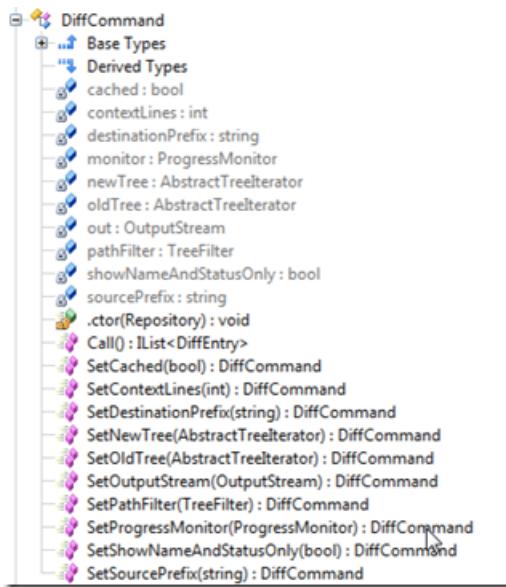
```
--DiffEntry[MODIFY testFile.bd]
```

Our objective is to get the Diff formatted output shown in the NGit Unit test.

A quick look at the Diff class, shows no public fields, properties or methods that expose it:

¹<http://blog.diniscruz.com/p/owasp-o2-platform.html>

²<http://blog.diniscruz.com/p/c-repl-script-environment.html>



And by default the *out* field is null:

The screenshot shows a debugger interface with two panes. The left pane, titled 'Command To Execute', contains the following JavaScript code:

```

1 var tempRepo = "tempRepo".tempDir();
2 var nGit = tempRepo.git_Init();
3
4 var file = "testFile.txt";
5 nGit.writeFile(file, "some content\naaaa\n");
6 nGit.add(".",false).commit_using_Status();
7 nGit.writeFile(file, "some content\naa Change\n");
8
9 var diff = nGit.Git.Diff();
10 diff.Call();
11 return diff.field("out");
12

```

The right pane, titled 'Invoke and Result', has two sections: 'Execute' and 'Output'. The 'Output' section displays the value '[null value]'.

Basically what we need to do is this:

The screenshot shows a debugger interface with two panes. The left pane, titled 'Command To Execute', contains C# code that attempts to create a `ByteArrayOutputStream` object. The right pane, titled 'Invoke and Result', shows the results of the execution, including a list of 'Compilation Errors'.

```

Command To Execute
1 var tempRepo = "tempRepo".tempDir();
2 var nGit = tempRepo.git_Init();
3
4 var file = "testFile.txt";
5 nGit.writeFile(file, "some content\naaaaa\n");
6 nGit.add(".", false).commit_using_Status();
7 nGit.writeFile(file, "some content\naa Change\n");
8
9 var diff = nGit.Git.Diff();
10 OutputStream @out = new ByteArrayOutputStream();
11 diff.SetOutputStream(@out);
12 diff.Call();
13 return diff.field("out");
14
15

```

Output

Compilation Errors:

- 35:25:CS0122::Sharpen.ByteArrayOutputStream' is inaccessible due to its protection level:c:\Users\o2\AppData\Local\Temp\3nmu24dz.0.cs
- 35:21:CS0143::The type 'Sharpen.ByteArrayOutputStream' has no constructors defined:c:\Users\o2\AppData\Local\Temp\3nmu24dz.0.cs

But as you can see, we can't create an instance of the `Sharpen.ByteArrayOutputStream`

Well, we can't create it directly, but we can easily create it using reflection :)

To do that, lets start by getting an reference to the `Sharpen.dll` assembly

The screenshot shows a debugger interface with two panes. The left pane, titled 'Command To Execute', contains C# code that uses reflection to get the assembly reference for `Sharpen.dll`. The right pane, titled 'Invoke and Result', shows the results of the execution, including a list of 'Output' details.

```

Command To Execute
1 return "Sharpen.dll".assembly();
2
3
4 var tempRepo = "tempRepo".tempDir();
5 var nGit = tempRepo.git_Init();
6
7 var file = "testFile.txt";
8 nGit.writeFile(file, "some content\naaaaa\n");
9 nGit.add(".", false).commit_using_Status();
10 nGit.writeFile(file, "some content\naa Change\n");
11
12 var diff = nGit.Git.Diff();
13 //OutputStream @out = new ByteArrayOutputStream();
14 //diff.SetOutputStream(@out);
15 diff.Call();

```

Output

Misc	
CodeBase	file:///E:/TeamMento
EntryPoint	
EscapedCode	file:///E:/TeamMento
Evidence	(Collection)
FullName	Sharpen, Version=1.0
GlobalAssem	False
HostContext	0

then add a reference to the `ByteArrayOutputStream` type

The screenshot shows a debugger interface with two main panes. The left pane, titled 'Command To Execute', contains the following C# code:

```

1 return "Sharpen.dll".assembly()
2     .type("ByteArrayOutputStream")
3
4
5 var tempRepo = "tempRepo".tempDir();
6 var nGit = tempRepo.git_Init();
7
8 var file = "testFile.txt";
9 nGit.writeFile(file, "some content\naaaa\n");
10 nGit.add(".",false).commit_using_Status();
11 nGit.writeFile(file, "some content\nnaa Change\n");
12
13 var diff = nGit.Git.Diff();
14 //OutputStream @out = new ByteArrayOutputStream();
15 //diff.SetOutputStream(@out);

```

The right pane, titled 'Invoke and Result', shows the results of the command. It includes an 'Output' section with the following table:

MemberType	TypeInfo
MetadataTokk	33554462
Module	Sharpen.dll
Name	ByteArrayOutputStrea
Namespace	Sharpen
ReflectedTyp	
StructLayout&	System.Runtime.InteropServices.StructLayoutAttribute
TypeHandle	System.RuntimeTypeHandle

invoke its constructor to create a live instance of it:

The screenshot shows a debugger interface with two main panes. The left pane, titled 'Command To Execute', contains the following C# code:

```

1 return "Sharpen.dll".assembly()
2     .type("ByteArrayOutputStream")
3     .ctor();
4
5
6 var tempRepo = "tempRepo".tempDir();
7 var nGit = tempRepo.git_Init();
8
9 var file = "testFile.txt";
10 nGit.writeFile(file, "some content\naaaa\n");
11 nGit.add(".",false).commit_using_Status();
12 nGit.writeFile(file, "some content\nnaa Change\n");
13
14 var diff = nGit.Git.Diff();
15 //OutputStream @out = new ByteArrayOutputStream();

```

The right pane, titled 'Invoke and Result', shows the results of the command. It includes an 'Output' section with the following table:

MemberType	TypeInfo
MetadataTokk	33554462
Module	Sharpen.dll
Name	ByteArrayOutputStrea
Namespace	Sharpen
ReflectedTyp	
StructLayout&	System.Runtime.InteropServices.StructLayoutAttribute
TypeHandle	System.RuntimeTypeHandle

since the Sharpen.OutputStream class is public, we can cast our *ByteArrayOutputStream* object into it:

The screenshot shows a debugger interface with two main panes. The left pane, titled 'Command To Execute', contains the following C# code:

```

1 return "Sharpen.dll".assembly()
2     .type("ByteArrayOutputStream")
3     .ctor()
4     .cast<OutputStream>();
5
6
7 var tempRepo = "tempRepo".tempDir();
8 var nGit = tempRepo.git_Init();
9
10 var file = "testFile.txt";
11 nGit.writeFile(file, "some content\naaaa\n");
12 nGit.add(".",false).commit_using_Status();
13 nGit.writeFile(file, "some content\nnaa Change\n");
14
15 var diff = nGit.Git.Diff();

```

The right pane, titled 'Invoke and Result', shows the results of the command. It includes an 'Output' section with the following table:

MemberType	TypeInfo
MetadataTokk	33554462
Module	Sharpen.dll
Name	ByteArrayOutputStrea
Namespace	Sharpen
ReflectedTyp	
StructLayout&	System.Runtime.InteropServices.StructLayoutAttribute
TypeHandle	System.RuntimeTypeHandle

we then assign it to the NGit command, which will give us the diff log we wanted

The screenshot shows a debugger interface with two main panes. The left pane, titled 'Command To Execute', contains the following C# code:

```

1 var outputStream = "Sharpen.dll".assembly()
2     .type("ByteArrayOutputStream")
3     .ctor()
4     .cast<OutputStream>();
5
6
7 var tempRepo = "tempRepo".tempDir();
8 var nGit = tempRepo.git_Init();
9
10 var file = "testFile.txt";
11 nGit.writeFile(file, "some content\naaaa\n");
12 nGit.add(".", false).commit_using_Status();
13 nGit.writeFile(file, "some content\nnaa Change\n");
14
15 var diff = nGit.Git.Diff();
16 //OutputStream @out = new ByteArrayOutputStream();
17 diff.SetOutputStream(outputStream);
18 diff.Call();
19 return outputStream.str();
20

```

The right pane, titled 'Invoke and Result', shows the output of the command. It includes a 'Execute' button and a 'stop execution' link. The 'Output' section displays the git diff output:

```

diff -git a/testFile.txt b/testFile.txt
index 1af3633..543e1d9 100644
--- a/testFile.txt
+++ b/testFile.txt
@@ -1,2 +1,2 @@
some content
aaaa
+aa Change

```

Note that the out field is now not null:

The screenshot shows a debugger interface with two main panes. The left pane, titled 'Command To Execute', contains the following C# code:

```

12 //OutputStream @out = new ByteArrayOutputStream();
13 nGit.writeFile(file, "some content\nnaa Change\n");
14
15 var diff = nGit.Git.Diff();
16 //OutputStream @out = new ByteArrayOutputStream();
17 diff.SetOutputStream(outputStream);
18 diff.Call();
19 //return outputStream.str();
20 return diff.field("out").str();
21
22
23

```

The right pane, titled 'Invoke and Result', shows the output of the command. It includes a 'Execute' button and a 'stop execution' link. The 'Output' section displays the git diff output:

```

diff -git a/testFile.txt b/testFile.txt
index 1af3633..543e1d9 100644
--- a/testFile.txt
+++ b/testFile.txt
@@ -1,2 +1,2 @@
some content
aaaa
+aa Change

```

Here is the Source code of the C# code snippet created:

```

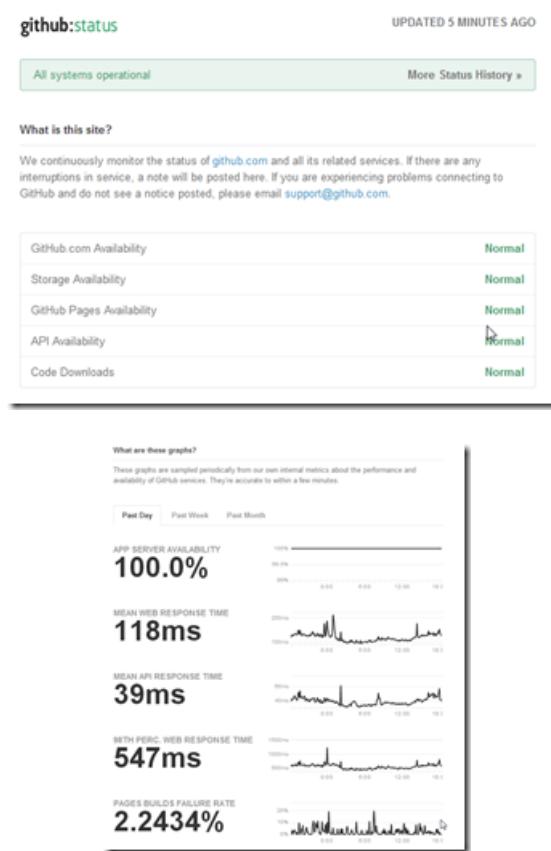
1 var outputStream = "Sharpen.dll".assembly()
2     .type("ByteArrayOutputStream")
3     .ctor()
4     .cast<OutputStream>();
5
6
7 var tempRepo = "tempRepo".tempDir();
8 var nGit = tempRepo.git_Init();
9
10 var file = "testFile.txt";
11 nGit.writeFile(file, "some content\naaaaa\n");
12 nGit.add(".", false).commit_using_Status();
13 nGit.writeFile(file, "some content\nnaa Change\n");

```

```
14
15 var diff = nGit.Git.Diff();
16
17 diff.SetOutputStream(outputStream).Call();
18 return outputStream.str();
19
20 //using Sharpen
21 //O2Ref:FluentSharp.NGit.DLL
22 //O2Ref:NGit.dll
23 //O2Ref:Sharpen.dll
```

GitHub.com needs to improve their the ‘Normal’ status definition and error reporting dashboard

At the moment (i.e. when I wrote this post), all should be ‘Normal’ with GitHub, since according to their status³, their world looks like this:



But to me here, a simple/small git push took about 5 minutes of retries:

³<https://status.github.com/>

```
02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git push origin
Connection closed by 207.97.227.239
fatal: The remote end hung up unexpectedly

02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git push origin
Read from socket failed: Connection reset by peer
fatal: The remote end hung up unexpectedly

02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git push origin
Connection closed by 207.97.227.239
fatal: The remote end hung up unexpectedly

02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ 
$ 

02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git push origin
ssh: connect to host github.com port 22: Bad file number
fatal: The remote end hung up unexpectedly

02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git push origin
Counting objects: 32, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (18/18), done.
Writing objects: 100% (18/18), 1.76 KiB, done.
Total 18 (delta 15), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
  0ef7358..5264b28  master -> master
```

which doesn't look 'Normal' to me!

I think GitHub.com needs to improve their the 'Normal' status definition and error reporting dashboard

Note that yesterday was the same thing.

Something weird is going on in GitHub's land

Another GitHub 'Normal' status that doesn't allow me to Push

It took me 5 attempts over about 10m (from 7:40pm on 11th Mar 2013) to get a commit pushed into GitHub's servers:

The screenshot shows two windows side-by-side. On the left is the GitHub status page at https://status.github.com/graphs/past_day. It displays a green bar indicating 'All systems operational' and a message 'UPDATED LESS THAN A MINUTE AGO'. Below this, there's a section titled 'What is this site?' with a note about monitoring GitHub and its services. On the right is a terminal window showing five consecutive failed attempts to push code to the 'origin' repository. Each attempt fails with the error 'ssh: connect to host github.com port 22: Bad file descriptor' and 'fatal: The remote end hung up unexpectedly'. The terminal session starts with 'MINGW32:/e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev\$ git push origin'.

It looks like GitHub is having DDos probs, which is another reason why they need to improve their visibility into what is going on:

github:status UPDATED ABOUT A MINUTE AGO

[« Dashboard](#)

Status Messages

Today

23:59 UTC All systems reporting at 100%

Yesterday, March 10, 2013

13:42 UTC Everything operating normally.

11:11 UTC Access to downloadable source code archives and uploaded files is temporarily down. We're working to restore it asap.

10:52 UTC We're recovering from another brief DDoS attack. Service should be returning to normal soon.

10:43 UTC Major service outage.

March 09, 2013

13:11 UTC Everything operating normally.

13:06 UTC We're recovering from what appears to be a DDoS attack. Services are returning to normal.

13:03 UTC Major service outage.

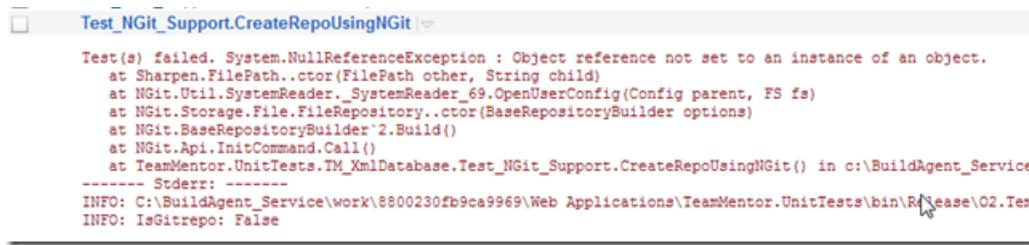
Prob with (older version of) NGit where it was failing to create Git repositories in Azure/TeamCity

Using an NGit version from a couple months ago.

```
1 var userHome = Path.Combine(Path.Combine(HostingEnvironment.ApplicationPhysicalPath, "App_Data\  
2 a"), "git");  
3 //using System.IO;  
4 //using System.Web.Hosting;  
5  
6 var runTime = "Sharpen.dll".assembly().type("Runtime");  
7  
8 var properties = (Hashtable)runTime.invokeStatic("GetProperties");  
9  
10 var result = "There are {0} properties <br>".format(properties.size());  
11 properties["user.home"] = userHome;  
12 //return properties["user.home"];  
13 foreach(DictionaryEntry item in properties)  
14     result+= "{0} = {1}<br/>".format(item.Key, item.Value);  
15  
16 return result;  
17  
18 var tempRepo = "tempRepo".tempDir(false);  
19 try  
20 {  
21     var initCommand = NGit.Api.Git.Init();  
22     initCommand.SetDirectory(tempRepo);  
23     initCommand.Call();  
24 }  
25 catch(Exception ex)  
26 {  
27     return ex.Message + ex.StackTrace;  
28 }  
29  
30 //O2Ref:NGit.dll  
31 //O2Ref:Sharpen.dll  
32 //return new API_NGit().init(tempRepo).str();  
33 return tempRepo.isGitRepository();  
34  
35  
36 //using O2.FluentSharp;  
37  
38 /*var runTime = "Sharpen.dll".assembly().type("Runtime");  
39 var properties = (Hashtable)runTime.invokeStatic("GetProperties");  
40 var result = "There are {0} properties <br>".format(properties.size());
```

```
41 foreach(DictionaryEntry item in properties)
42     result+= "{0} = {1}<br/>".format(item.Key, item.Value);
43 
44 return result;
45 return "FluentSharp.NGit.dll".assembly();*/
46 
47 //using System.Collections;
48 //O2Ref:TeamMentor.CoreLib.dll
49 //O2Ref:FluentSharp.NGit.dll
```

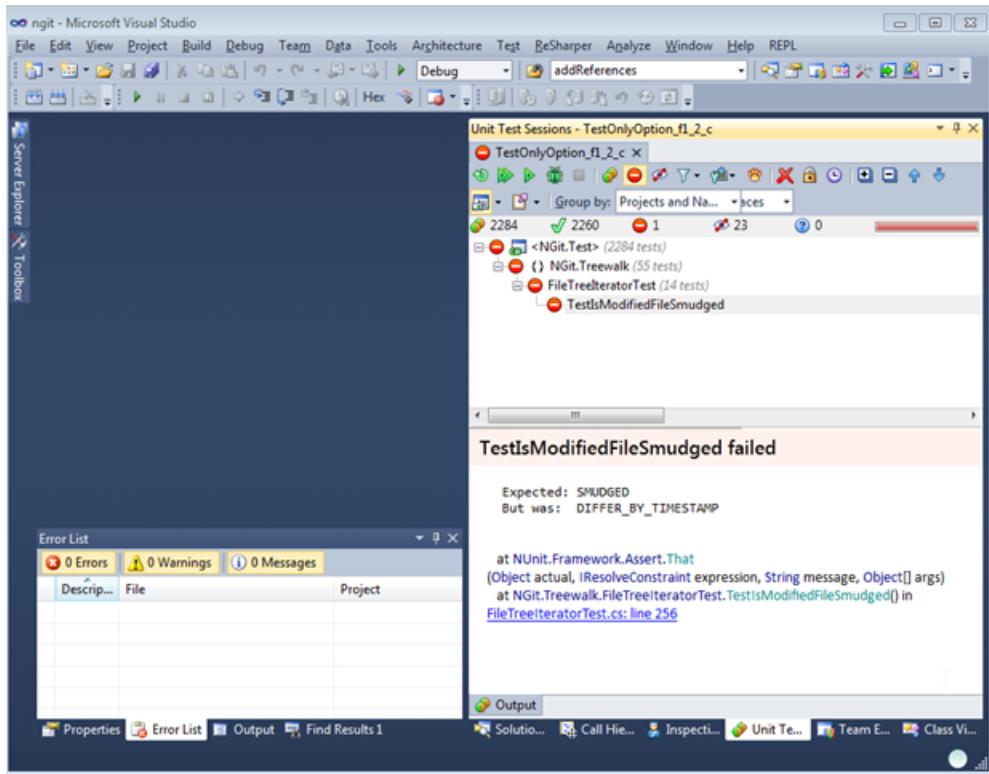
I would get this exception:



The error is the one documented here: <https://github.com/mono/ngit/issues/14>⁴

After forking the repo and building it locally in VS 2010, I run the UnitTests (note: only one of 2284 tests failed after a couple retries (there were about 20 that failed on first execution, but passed to individual retest)):

⁴<https://github.com/mono/ngit/issues/14>



I then added the compiled assemblies to TeamMentor, and now this script:

```

1 var properties = Sharpen.Runtime.GetProperties();
2 var result = "There are {0} properties <br>".format(properties.size());
3 //properties["user.home"] = userHome;
4 //return properties["user.home"];
5 foreach(DictionaryEntry item in properties)
6     result+= "{0} = {1}<br/>".format(item.Key, item.Value);
7
8
9 var tempRepo = "tempRepo".tempDir(false);
10 try
11 {
12     var initCommand = NGit.Api.Git.Init();
13     initCommand.SetDirectory(tempRepo);
14     initCommand.Call();
15 }
16 catch(Exception ex)
17 {

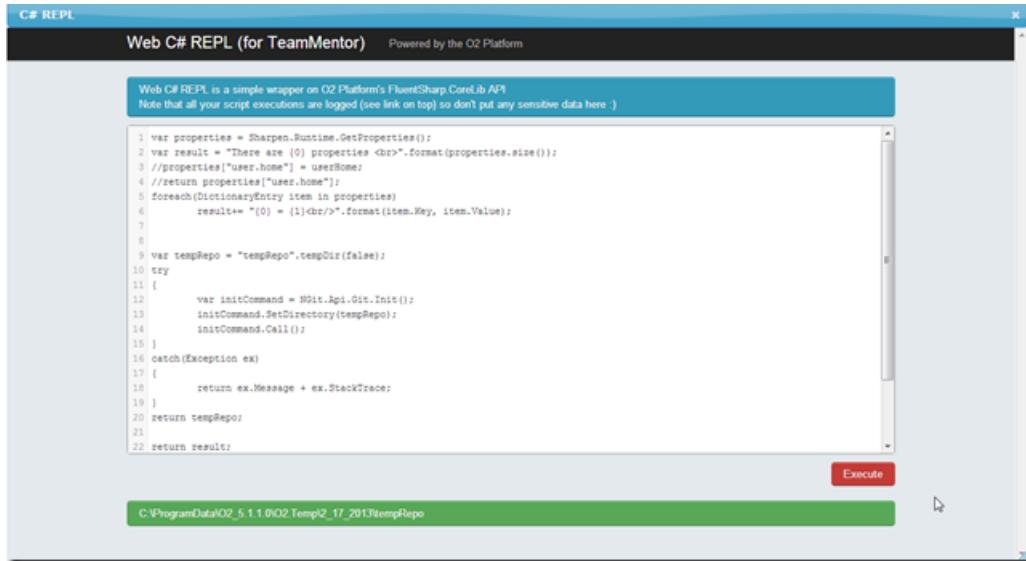
```

```

18     return ex.Message + ex.StackTrace;
19 }
20 return tempRepo;
21
22 return result;
23
24
25 //using System.Collections;
26 //O2Ref:Sharpen.dll
27 //O2Ref:NGit.dll

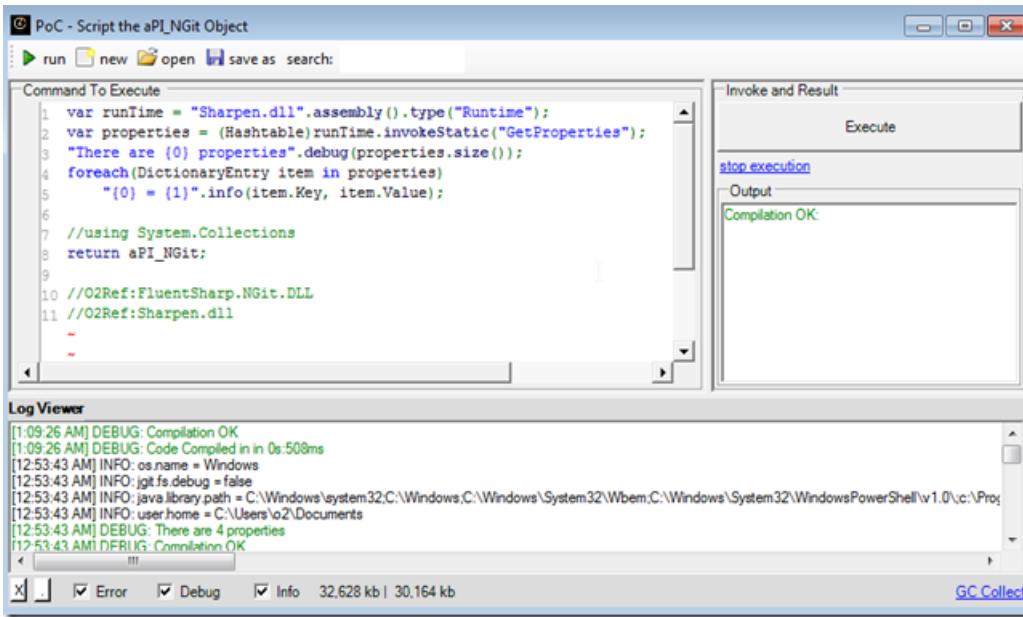
```

Executed on the O2 Platform Browser-based C# REPL:



Creates the local repository on a local folder.

NOTE: For reference here is how to write that reflection script using the O2 API;s



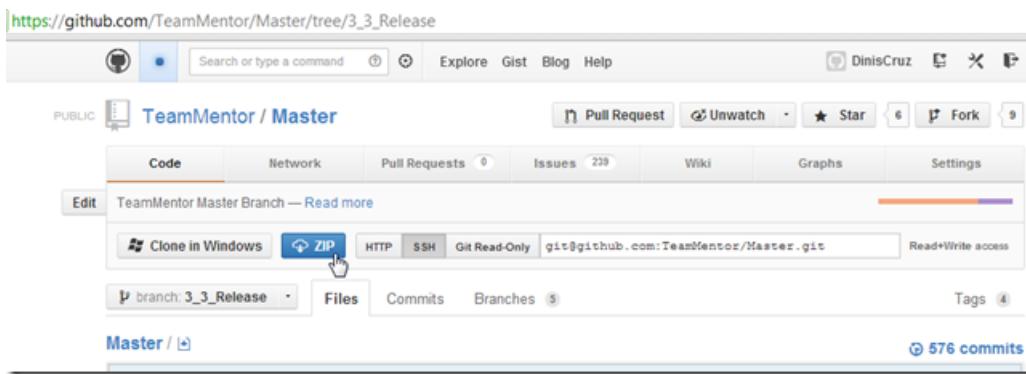
```
1 var runTime = "Sharpen.dll".assembly().type("Runtime");
2 var properties = (Hashtable)runTime.invokeStatic("GetProperties");
3 "There are {0} properties".debug(properties.size());
4 foreach(DictionaryEntry item in properties)
5     "{0} = {1}".info(item.Key, item.Value);
6
7 //using System.Collections
8 return aPI_NGit;
9
10 //O2Ref:FluentSharp.NGIt.DLL
11 //O2Ref:Sharpen.dll
```

Creating a version TeamMentor which uses the new GitUserData.config file

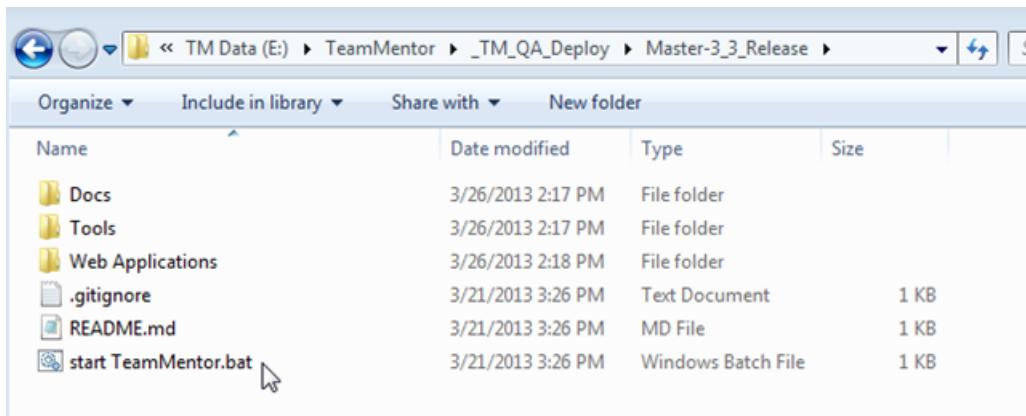
Introduced in the 3.3 version of TM is a new feature to load the UserData repository from an external location (GitHub or local folder).

This post shows how to set it up.

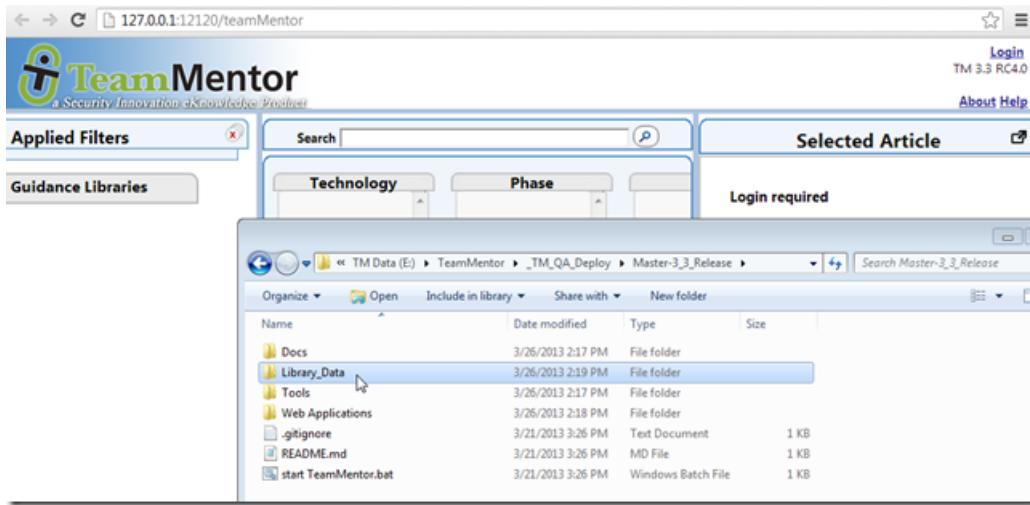
First step is to get the latest version of TeamMentor from GitHub, where we can clone it locally or download the zip file



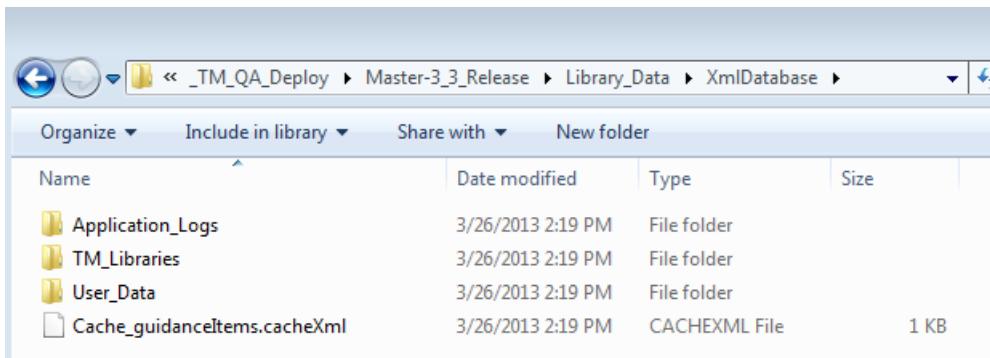
Using the Zip file has an example, unzip the 10Mb file into a local folder, and click on *start_TeamMentor.bat*



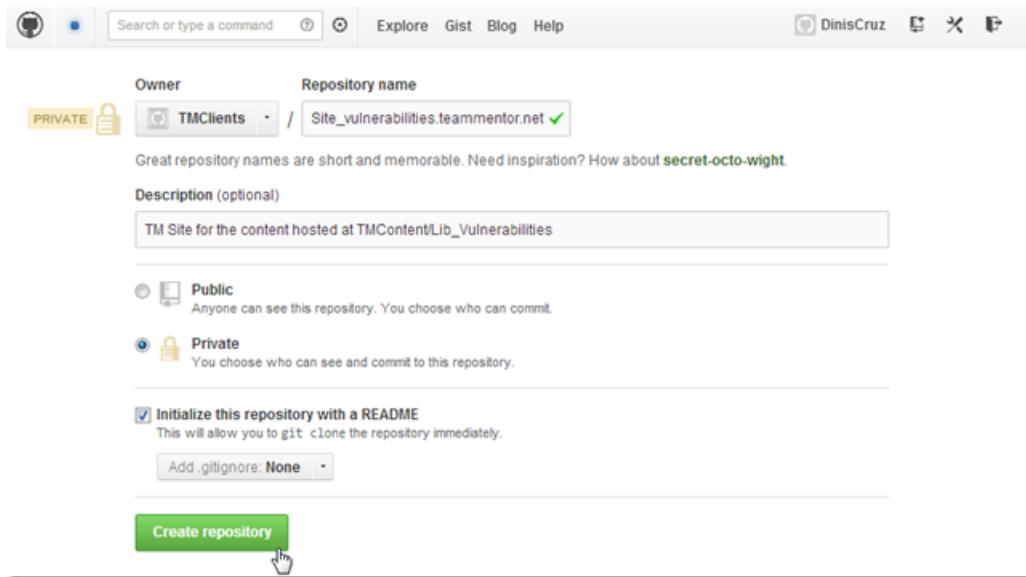
This will open an empty TM site, and a new Library_Data folder should have been created:



With this default structure:



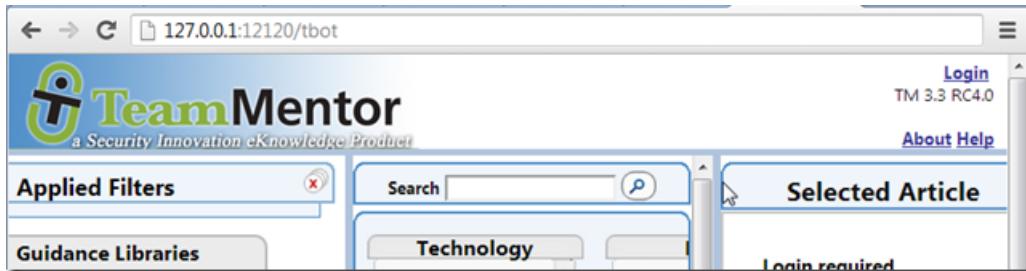
Now in GitHub (or on a local folder), create a new Git Repository (which should be marked as private, since security sensitive data will be stored here)



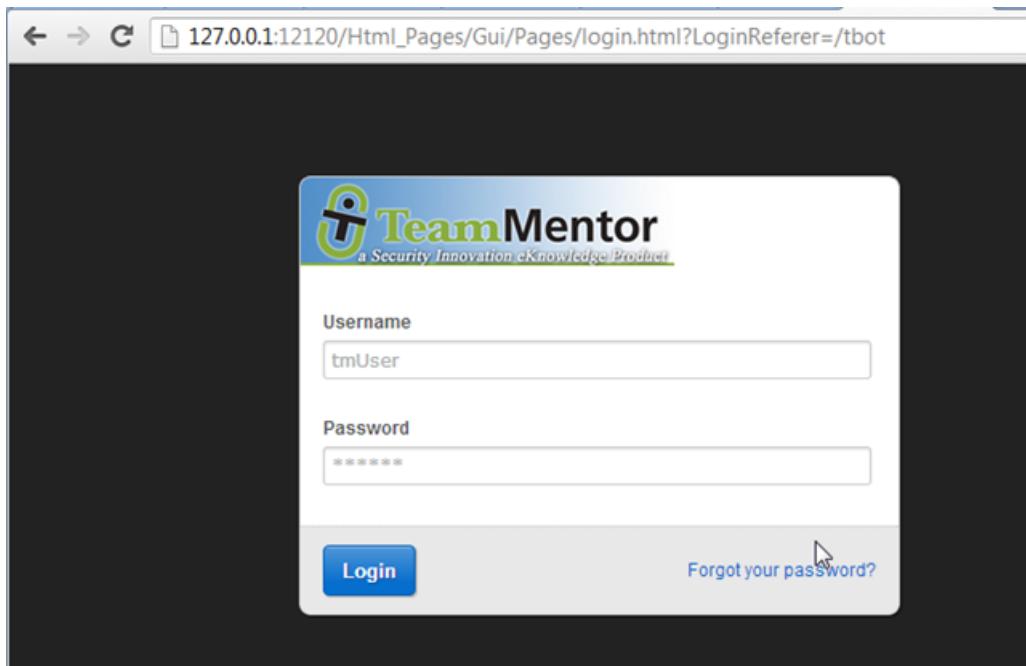
Once the repository is created, copy its git url (in this case git@github.com:TMClients/Site_vulnerabilities.teammentor.net.git⁵)

⁵mailto:git@github.com:TMClients/Site_vulnerabilities.teammentor.net.git

Back in the local copy of TeamMentor, open the TBot page:



which will require an admin account:



After login, open the *Edit GitUserLocation*

A screenshot of a web browser window titled "localhost:12120/rest/tbot/run/Commands". The page header says "TBot - your friendly TeamMentor Bot". Below it, the section title "Available TBot Commands" is displayed. A sub-instruction reads "Welcome to the TBot control center, please chose the command to execute:". A grid of command links is shown:

Admin	Import Legacy Users	Reload Server Objects		
Admin\Config	Edit GitUserLocation	Edit SecretData	View SecretData	View TMConfig
Admin\LiveData	ActiveSessions	DebugInfo	IISSessions	
LibraryData	Articles	Libraries Xml		

And enter the Git url copied from GitHub:

A screenshot of a web browser window titled "localhost:12120/rest/tbot/run/Edit_GitUserLocation". The page header says "TBot - your friendly TeamMentor Bot". The section title "Editing an Secret Data" is displayed. A form field labeled "Git User Location:" contains the value "git@github.com:TMClients/Site_vulnerabilities.teammendor.net.git". Below the form is a blue "Save" button with the text "data loaded". At the bottom, a link "back to commands list" is visible.

After the data is saved, go back to the commands list:

A screenshot of a web browser window showing the "commands list" page. It displays the same "Edit GitUserLocation" form and "Save" button as the previous screenshot, but with the message "data saved: true" below the button. At the bottom, a link "back to commands list" is visible.

Go to the Reload Server Objects

A screenshot of a web browser window titled "localhost:12120/rest/tbot/run/Commands". The page header says "TBot - your friendly TeamMentor Bot". Below it, a section titled "Available TBot Commands" contains the following buttons:

Admin	Import Legacy Users	Reload Server Objects
AdminConfig	Edit GitUserLocation	Edit SecretData
		View SecretData
		View TMConfig

The "Reload Server Objects" button is highlighted with a cursor pointing at it.

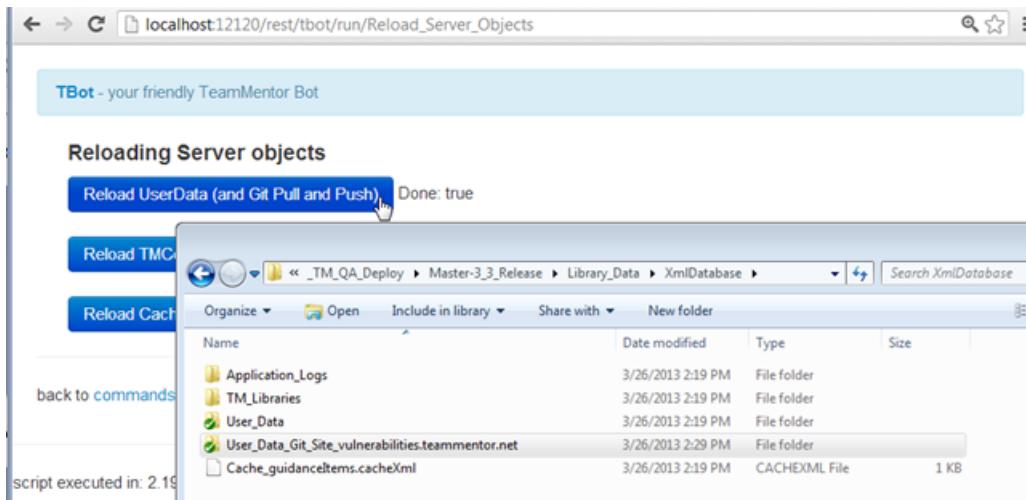
And click on 'Reload UserData':

A screenshot of a web browser window titled "localhost:12120/rest/tbot/run/Reload_Server_Objects". The page header says "TBot - your friendly TeamMentor Bot". Below it, a section titled "Reloading Server objects" contains the following buttons:

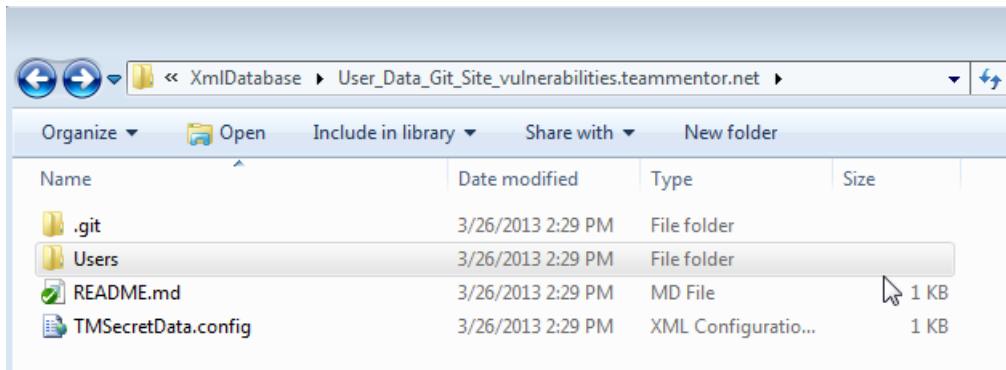
Reload UserData (and Git Pull and Push)
Reload TMConfig
Reload Cache (will reload entire Xml database, including User data)

The "Reload UserData (and Git Pull and Push)" button is highlighted with a cursor pointing at it.

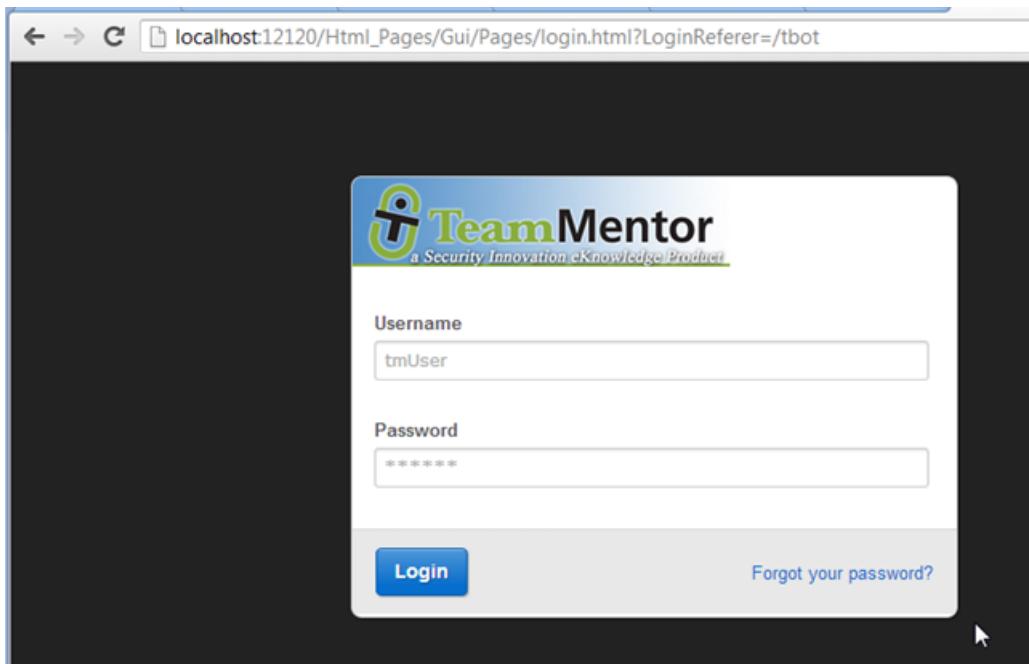
After that step is completed, if you look at the *Library* Data folder, you should see a new UserData folder in there (that uses the git repository name as part of its path)



Inside it, you will see the README.md that was received from GitHub, and a new TMSecretData.config and Users folder



Back in TBot's page, if you click on any link you should be redirected to the login page, and you will need to login again using the default admin credentials (this happens because the current browser cookie is pointing to the admin user that is in **XmlDatabaseUser_Data** and not in the newly created **XmlDatabaseUser_Data_Git_Site_vulnerabilities.teammendor.net**:



After logging in, open the *Edit SecretData* command:

A screenshot of a web browser window. The address bar shows the URL: localhost:12120/rest/tbot/run/Commands. The page title is 'TBot - your friendly TeamMentor Bot'. Below the title, it says 'Available TBot Commands' and 'Welcome to the TBot control center, please chose the command to execute:'. There is a grid of command links:

Admin	Import Legacy Users	Reload Server Objects
Admin\Config	Edit GitUserLocation	Edit SecretData
Admin\LiveData	ActiveSessions	DebugInfo
LibraryData	Articles	Libraries Xml

The 'Edit SecretData' link in the second row is highlighted with a mouse cursor. The background of the page is white, and the overall layout is clean and modern.

Which should look like this (with correct values for the Rijndael and SMTP fields):

```
{
  "Libraries_Git.Repositories": [],
  "Rijndael_IV": "AAAAAAAAAAAAAAAL9pvKLHvQ==",
  "Rijndael_Key": "AAAAAAAAAAAAAA/dWVoGX5dNy+YDM9j2tvY=",
  "SMTP_Password": null,
  "SMTP_Server": "BBBBBB",
  "SMTP_UserName": "CCCCCC"
}
```

Save data loaded

The value that we want to change is the ***Libraries_Git.Repositories***, which should point to the Git repo we want to add to this TM instance. In this case:

TMContent / Lib_Vulnerabilities

No description or homepage.

git@github.com:TMContent/Lib_Vulnerabilities Read+Write access

branch: master

File	Changed	Author	Date	Commit Message
Articles	6 days ago	unknown-user	6 days ago	changed: Articles/ee8e0bbc-09bc-49fb-9690-2a8ecbd64916.xml
_GuidanceItems	7 days ago			Renaming Articles due to GUID conflict with .NET 4.0 library [DinisCruz]
README.md	4 months ago			Initial commit [SergeTruth]
Vulnerabilities.xml	7 days ago			Added: Articles/ee8e0bbc-09bc-49fb-9690-2a8ecbd64916.xml [DinisCruz]

Add the git url as an item in the ***Libraries_Git.Repositories*** Javascript array:

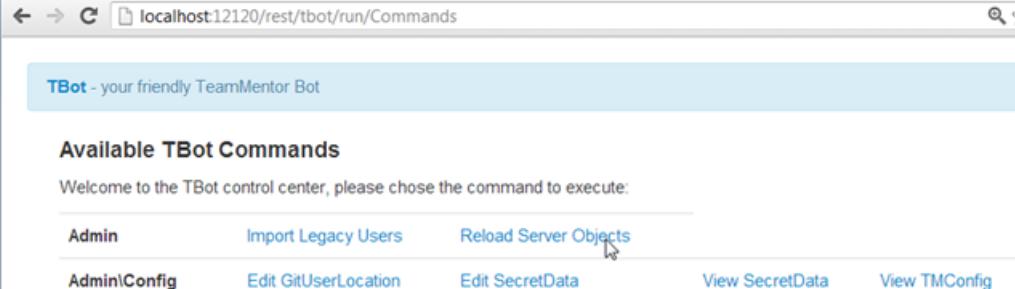


A screenshot of a JSON editor interface. The code shown is:

```
{  
    "Libraries_Git.Repositories": [  
        {"git@github.com:TMContent/Lib_Vulnerabilities.git"},  
        {"Rijndael_IV": "AAAAAAAAAAAAAAAAL9pvKLHvQ==", "Rijndael_Key": "AAAAAAAAAAAAAAAAdVVoGX5dNy+YDM9j2tvY=", "SMTP_Password": null, "SMTP_Server": "BBBBBB", "SMTP_UserName": "CCCCCC"}  
    ]  
}
```

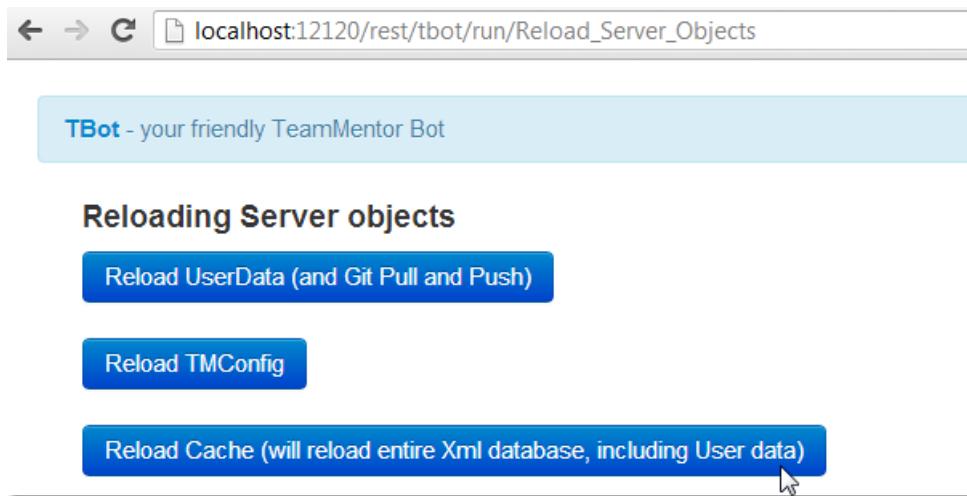
A blue 'Save' button with a hand cursor icon is visible on the right side of the editor window. A tooltip 'data loaded' is also present near the button.

After the data is saved, open the *Reload Server Objects* again:



A screenshot of a web browser window titled 'localhost:12120/rest/tbot/run/Commands'. The page header says 'TBot - your friendly TeamMentor Bot'. Below it, a section titled 'Available TBot Commands' contains the text 'Welcome to the TBot control center, please chose the command to execute:'. A horizontal menu bar has several items: Admin, Import Legacy Users, Reload Server Objects (which is highlighted with a mouse cursor), AdminConfig, Edit GitUserLocation, Edit SecretData, View SecretData, and View TMConfig.

And this time around click on the *Reload Cache* button:



Once that is completed, if you open the ***XmldbDatabase/TM_Libraries*** folder, you should see a new ***Vulnerabilities*** subfolder

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, including User data) Done: "In the library 'E:\TeamMentor_TM_QA_Deploy\Master-3_3_Release\Library_Data\XmlDatabase\TM_Libraries' there are 1 library(ies), 1 views and 64 Guidanceitems"

back script ex

Name	Date modified	Type	Size
Vulnerabilities	3/26/2013 2:42 PM	File folder	

Which is in fact a git clone:

Name	Date modified	Type	Size
.git	3/26/2013 2:42 PM	File folder	
_GuidanceItems	3/26/2013 2:42 PM	File folder	
Articles	3/26/2013 2:42 PM	File folder	
README.md	3/26/2013 2:42 PM	MD File	1 KB
Vulnerabilities.xml	3/26/2013 2:42 PM	XML File	1 KB

of the git repository configured on the **Libraries_Git.Repositories** value

```

MINGW32:/e/TeamMentor/_TM_QA_Deploy/Master-3_3_Release/Library_Data/XmlDatabase/TM_Libraries/Vulnerabilities
Welcome to Git (version 1.7.10-preview20120409)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

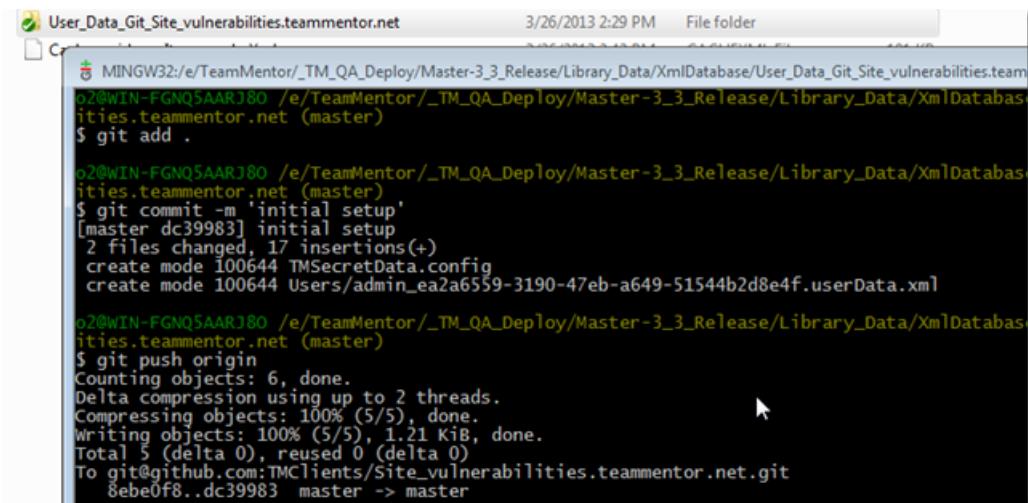
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_QA_Deploy/Master-3_3_Release/Library_Data/Xml
(master)
$ git remote -v
origin  git@github.com:TMContent/Lib_Vulnerabilities.git (fetch)
origin  git@github.com:TMContent/Lib_Vulnerabilities.git (push)

```

Quickly opening the main TM page, will now show the **Vulnerabilities** Library:

Title	Technology	Phase	Type
Client-Side Only Validation	ASP.NET 4.0	Implementation	Landi Page
Code Injection	PHP	Implementation	Landi Page

Final step is to do manually commit the changes made to the **User_Data_Git_Site_vulnerabilities.teammendor.net** local repository (note: auto commit and push is disabled on the UserData when running TM from localhost)



```

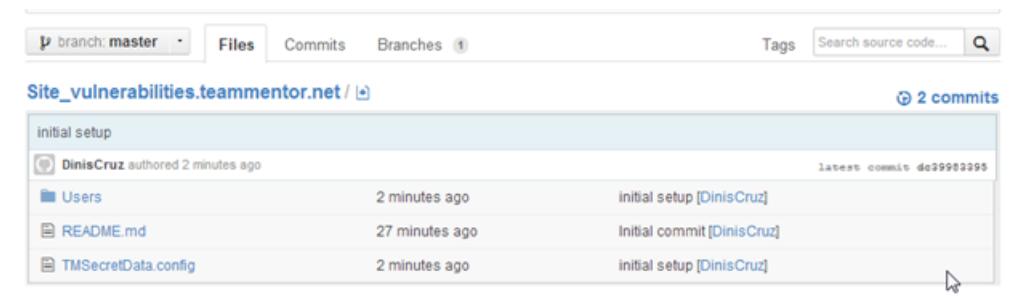
User_Data_Git_Site_vulnerabilities.teammendor.net      3/26/2013 2:29 PM  File folder
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_QA_Deploy/Master-3_3_Release/Library_Data/XmlDatabase/User_Data_Git_Site_vulnerabilities.team
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_QA_Deploy/Master-3_3_Release/Library_Data/XmlDatabase/User_Data_Git_Site_vulnerabilities.teammendor.net (master)
$ git add .

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_QA_Deploy/Master-3_3_Release/Library_Data/XmlDatabase/User_Data_Git_Site_vulnerabilities.teammendor.net (master)
$ git commit -m 'initial setup'
[master dc39983] initial setup
 2 files changed, 17 insertions(+)
 create mode 100644 TMSecretData.config
 create mode 100644 Users/admin_ea2a6559-3190-47eb-a649-51544b2d8e4f.userData.xml

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_QA_Deploy/Master-3_3_Release/Library_Data/XmlDatabase/User_Data_Git_Site_vulnerabilities.teammendor.net (master)
$ git push origin
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.21 KiB, done.
Total 5 (delta 0), reused 0 (delta 0)
To git@github.com:TMClients/Site_vulnerabilities.teammendor.net.git
 8bebe0f8..dc39983  master -> master

```

Which will put those updates in GitHub

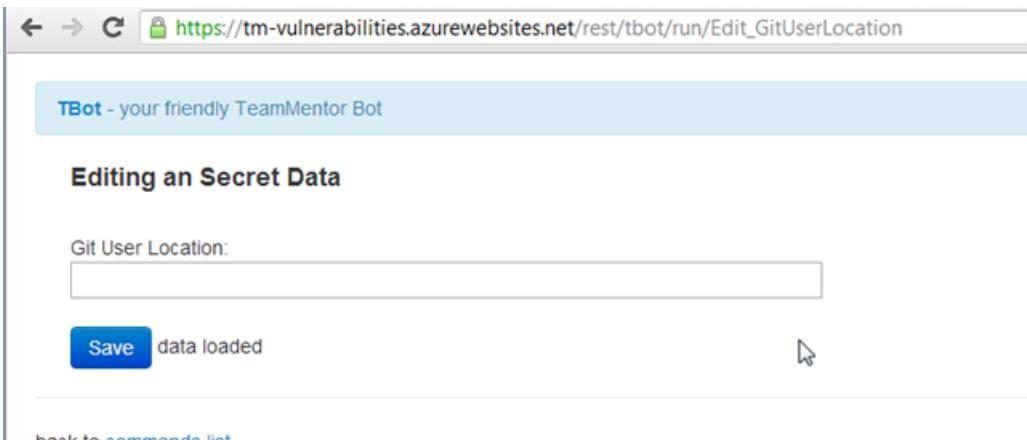


Site_vulnerabilities.teammendor.net / 2 commits

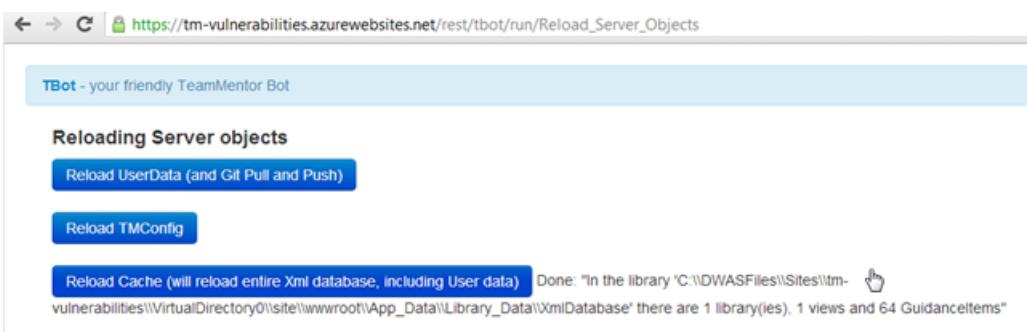
initial setup

	DinisCruz authored 2 minutes ago		latest commit dc39983
	Users	2 minutes ago	initial setup [DinisCruz]
	README.md	27 minutes ago	Initial commit [DinisCruz]
	TMSecretData.config	2 minutes ago	initial setup [DinisCruz]

Now that we have this GitHub repository configured, we can configure the **Git UserLocation** of live QA server:



And after reloading the cache:



The <https://tm-vulnerabilities.azurewebsites.net>⁶ will now have the ***Lib_Vulnerabilities*** library

⁶<https://tm-vulnerabilities.azurewebsites.net/>

The screenshot shows the TeamMentor application interface. At the top, there is a header with the TeamMentor logo, a security innovation eKnowledge Product, and navigation links for Edit Mode, Change Password, Control Panel, Logout, TM 3.3 RC4.0, and a user logged in as admin. Below the header, there are sections for Applied Filters, Guidance Libraries (Vulnerabilities), and a search bar. The main content area displays a search results grid for 63 items. The columns are Title, Technology, Phase, Type, and Category. The first three rows of the grid are:

Title	Technology	Phase	Type	Category
Client-Side Only Validation	ASP.NET 4.0	Implementation	Landing Page	Input Validation
Code Injection	PHP	Implementation	Landing Page	Input Validation
Code Injection	ASP.NET 4.0	Implementation	Landing Page	Dangerous

To the right, a selected article titled "Command Injection" is shown with a detailed description.

Selected Article

Command Injection

Description

Applications often execute external commands as a part of their functionality. If the attacker is able to manipulate the choice of external commands or their parameters, the attacker may be able to use this functionality to execute arbitrary commands. If an attacker is able to execute arbitrary commands, he is typically able to take over the application and possibly take over the entire system. All applications that use external applications are affected by command injection vulnerabilities. Web applications are especially at risk, because they are exposed to the Internet. To prevent command injection vulnerabilities, validate all

Changing the 'View TM article by anonymous users' status via GitHub

From the 3.3. release of TeamMentor (TM) it is now possible to change configuration settings of live servers directly from GitHub.

For example I just published a QA version of the <https://services.teammentor.net>⁷ site on Azure's <https://tm-services.azurewebsites.net>⁸

Here is what <https://services.teammentor.net>⁹ (on version 3.2.3) looks like:

The screenshot shows a web browser displaying the Team Mentor website at <https://services.teammentor.net>. The page title is "Team Mentor". On the left, there is a sidebar titled "Applied Filters" and "Guidance Libraries" containing a list of 20 vulnerability categories. The main content area has sections for "Technology" and "Phase" filters, both of which are currently set to "Any". Below these filters, a table lists 118 items. The first item in the list is "Allocate Enough Buffer Space for Copying Data", which is highlighted with a blue background. To the right of the table, a detailed view of this article is shown under the heading "Selected Guidance Item". The article title is "Allocate Enough Buffer Space for Copying Data". Under "What to Do", it says: "Only copy the amount of data that can fit in the array or string. Make sure that the array or string is large enough to fit all the data, including the possible trailing null." Under "Why", it says: "Copying more data than an array or string can fit results in a buffer overflow." Under "When", it says: "Allocate enough buffer space for copying data, when data is being copied to an array or a string."

Here is what <https://tm-services.azurewebsites.net>¹⁰ (on version 3.3 RC4) looks like:

⁷<https://services.teammentor.net/>

⁸[https://tm-services.azurewebsites.net/](https://tm-services.azurewebsites.net)

⁹<https://services.teammentor.net/>

¹⁰[https://tm-services.azurewebsites.net/](https://tm-services.azurewebsites.net)

The screenshot shows the TeamMentor website at <https://tm-services.azurewebsites.net/teamMentor#>. The interface includes a sidebar with 'Applied Filters' and 'Guidance Libraries' containing 20 vulnerability categories. A central search panel with 'Technology' and 'Phase' filters displays 114 items. On the right, a 'Selected Article' panel prompts for login with fields for Username, Password, Repeat Password, E-Mail, First Name, Last Name, Company, Title/Job, Country, and State, plus a 'Sign Up' button.

Can you spot the difference?

Here is the file (on GitHub) that controls if Anonymous users should be able to see TM's articles:

[Site_services.teammendor.net / TMConfig.config](#)

The screenshot shows the TMConfig.config file on GitHub. It contains XML configuration code with line numbers 1 through 13. Line 7 contains the key line: <Show_ContentToAnonymousUsers>false</Show_ContentToAnonymousUsers>. The GitHub interface shows 1 contributor and various file statistics.

```

1  <?xml version="1.0"?>
2  <TMConfig>
3    <TNSetup>
4      <XmlLibrariesPath>TM_Libraries_Services</XmlLibrariesPath>
5    </TNSetup>
6    <TNSecurity>
7      <Show_ContentToAnonymousUsers>false</Show_ContentToAnonymousUsers>
8    </TNSecurity>
9    <Git>
10      <AutoCommit_UserData>true</AutoCommit_UserData>
11      <AutoCommit_LibraryData>true</AutoCommit_LibraryData>
12    </Git>
13  </TMConfig>

```

So the solution is to edit this file (in GitHub):

[Site_services.teammendor.net / TMConfig.config](#)

DinisCruz 2 hours ago first commit of config files and admin user
1 contributor

file | 14 lines (13 sloc) | 0.377 kb

Edit Raw Blame History

```
<?xml version="1.0"?>
<TMConfig>
  <TMSetup>
    <XmlLibrariesPath>TM_Libraries_Services</XmlLibrariesPath>
  </TMSetup>
  <TMSecurity>
    <Show_ContentToAnonymousUsers>false</Show_ContentToAnonymousUsers>
  </TMSecurity>
  <Git>
    <AutoCommit_UserData>true</AutoCommit_UserData>
    <AutoCommit_LibraryData>true</AutoCommit_LibraryData>
  </Git>
</TMConfig>
```

change that value to true:

Site_services.teammendor.net / TMConfig.config or cancel

Code Preview

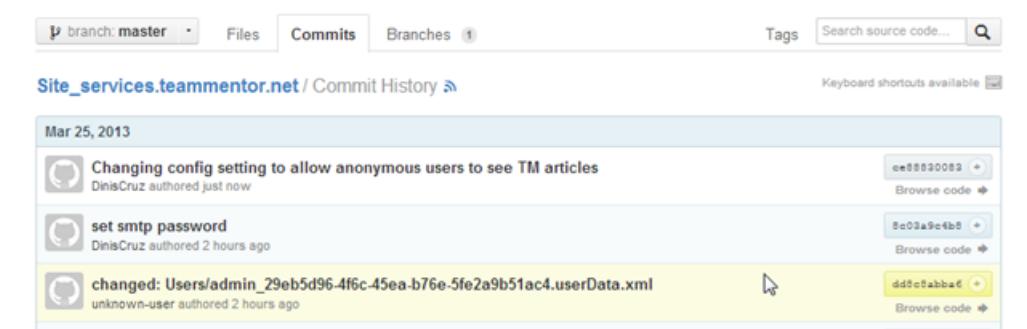
Spaces 2 No wrap

```
<?xml version="1.0"?>
<TMConfig>
  <TMSetup>
    <XmlLibrariesPath>TM_Libraries_Services</XmlLibrariesPath>
  </TMSetup>
  <TMSecurity>
    <Show_ContentToAnonymousUsers>true</Show_ContentToAnonymousUsers>
  </TMSecurity>
  <Git>
    <AutoCommit_UserData>true</AutoCommit_UserData>
    <AutoCommit_LibraryData>true</AutoCommit_LibraryData>
  </Git>
</TMConfig>
```

Commit that change in GitHub's UI:

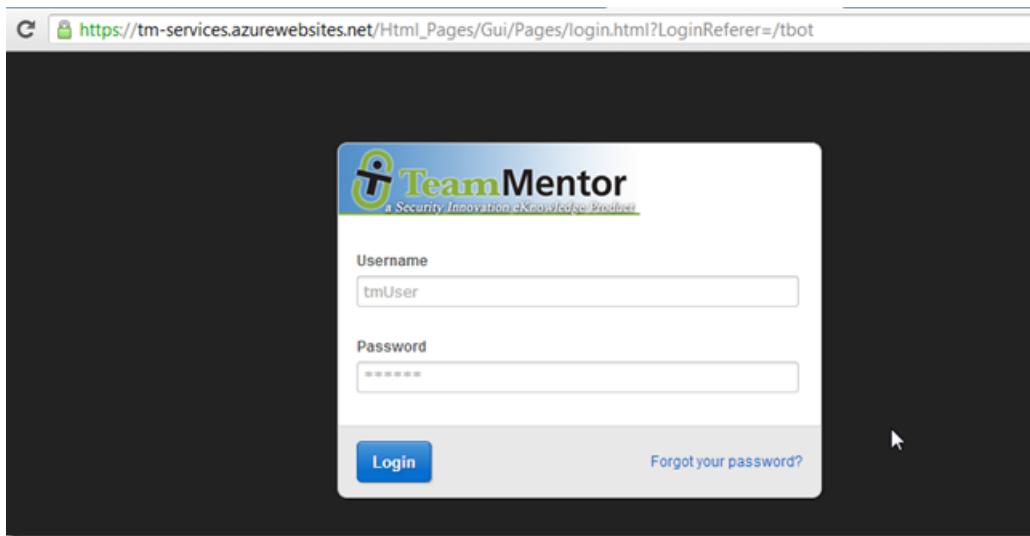


With this commit being now part of this repository:

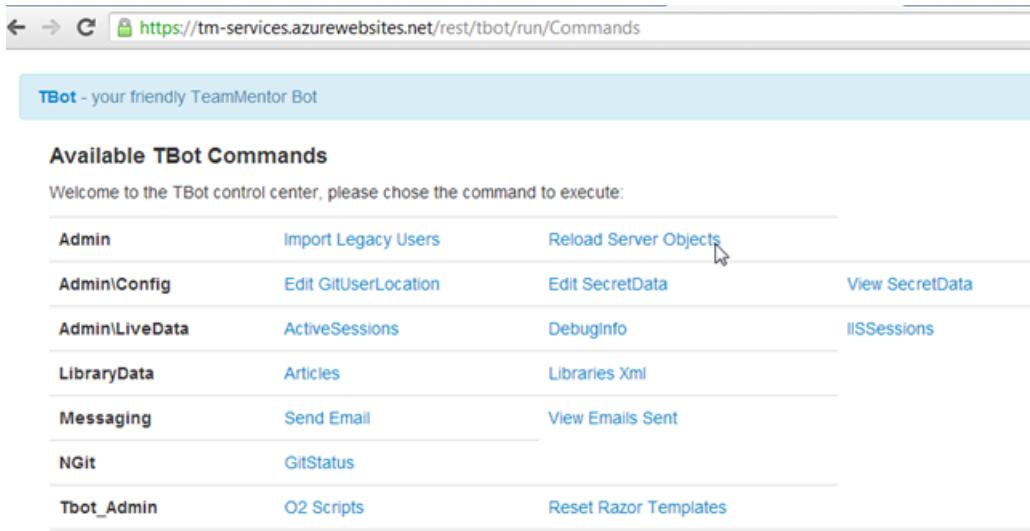


Next we go into the new Tbot interface (<https://tm-services.azurewebsites.net/tbot>¹¹), which requires admin privs:

¹¹<https://tm-services.azurewebsites.net/tbot>



After login , open the '*Reload Server Objects*' page



Available TBot Commands

Welcome to the TBot control center, please chose the command to execute:

Admin	Import Legacy Users	Reload Server Objects	
Admin\Config	Edit GitUserLocation	Edit SecretData	View SecretData
Admin\LiveData	ActiveSessions	DebugInfo	IISSessions
LibraryData	Articles	Libraries Xml	
Messaging	Send Email	View Emails Sent	
NGit	GitStatus		
Tbot_Admin	O2 Scripts	Reset Razor Templates	

And click on the *Reload UserData (and Git Pull and Push)* button

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, including User data)

[back to commands list](#)

script executed in: 0.4843626s

which when executed:

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push) Done: true

will have updated the local TMConfig.config file:

The screenshot shows a browser window with the URL https://tm-services.azurewebsites.net/rest/tbot/run/View_TMConfig. The page title is "TBot - your friendly TeamMentor Bot". Below it, the section title is "TMConfig.Config". The content is a block of XML code:

```

<?xml version="1.0"?>
<TMConfig>
  <TMSetup>
    <UseAppDataFolder>false</UseAppDataFolder>
    <TMLibraryDataVirtualPath>...</TMLibraryDataVirtualPath>
    <XmlLibrariesPath>TM_Libraries_Services</XmlLibrariesPath>
    <UserDataPath>User_Data</UserDataPath>
    <LibrariesUploadedFiles>LibrariesUploadedFiles</LibrariesUploadedFiles>
    <EnableGZipForWebServices>true</EnableGZipForWebServices>
    <Enable302Redirects>true</Enable302Redirects>
  </TMSetup>
  <TMSecurity>
    <Show_ContentToAnonymousUsers>true</Show_ContentToAnonymousUsers>
    <SSL_RedirectHttpToHttps>true</SSL_RedirectHttpToHttps>
    <EvalAccounts_Enabled>true</EvalAccounts_Enabled>
  
```

And if we logout, we will see the expected behavior:

The screenshot shows the TeamMentor web application interface. The top navigation bar includes "Login", "TM 3.3 RC4.0", and "About Help". The main content area has sections for "Applied Filters", "Search", and "Selected Article".

Applied Filters:

- Guidance Libraries
 - Top Vulnerabilities
 - 01 XSS
 - 02 SQLi
 - 03 CSRF
 - 04 Buffer Overflow
 - 05 Format String
 - 06 XML Injection
 - 07 SSL Misconfiguration
 - 08 Session Hijacking
 - 09 Weak Cryptography
 - 10 Information Disclosure
 - 11 Direct Request
 - 12 Forceful Browsing
 - 13 URL Redirection
 - 14 Unsecure Cookies
 - 15 Poor Password Storage
 - 16 Authentication Issues
 - 17 Key Mismangement
 - 18 Single Sign-on Issues
 - 19 Misc Command Injection
 - 20 Access Control Issues

Selected Article:

Allocate Enough Buffer Space for Copying Data

What to Do:

Only copy the amount of data that can fit in the array or string. Make sure that the array or string is large enough to fit all the data, including the possible trailing null.

Why:

Copying more data than an array or string can fit results in a buffer overflow.

When:

Allocate enough buffer space for copying data, when data is being copied to an array or a string.

Finally, if we look at GitHub's commit history, we will see the commit we did in GitHub

nicely merged with the commits that happened at the live server

The screenshot shows the GitHub commit history for the repository `Site_services.teammendor.net`. The commits listed are:

- Merge branch 'master' of https://tm-bot%21%21tbot%21%2112345@github.... (unknown-user authored 2 minutes ago)
- changed: Users/admin_29eb5d96-4f6c-45ea-b76e-5fe2a9b51ac4.userData.xml (unknown-user authored 3 minutes ago)
- Changing config setting to allow anonymous users to see TM articles (DinsCruz authored 5 minutes ago) - This commit is highlighted with a mouse cursor.
- changed: Users/admin_29eb5d96-4f6c-45ea-b76e-5fe2a9b51ac4.userData.xml (unknown-user authored 11 minutes ago)

Here is the GitHub's Network Graph of this repository, which shows both types of commits (the ones performed at at the live server vs the one at GitHub)

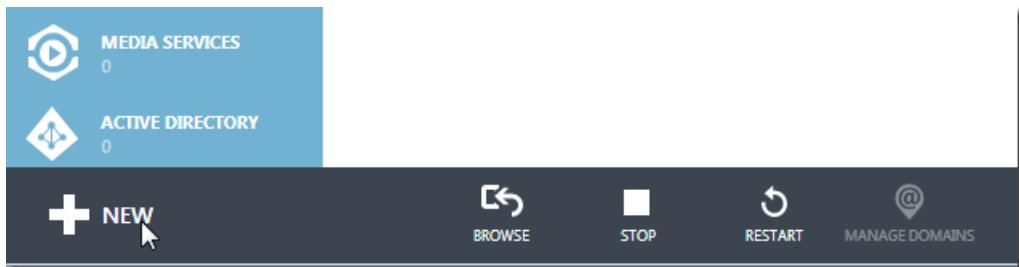
The screenshot shows the GitHub Network Graph for the repository `Site_services.teammendor.net`. The graph displays the network of branches connected to the reference point `TMClients/Site_services.teammendor.net`. The graph interface includes tabs for `Code`, `Network` (which is selected), `Pull Requests`, `Issues`, `Wiki`, `Graphs`, and `Settings`. The graph visualization shows several nodes connected by lines, with a prominent blue line representing the main connection path.

Creating a new TeamMentor test site using TeamCity, GitHub and Azure

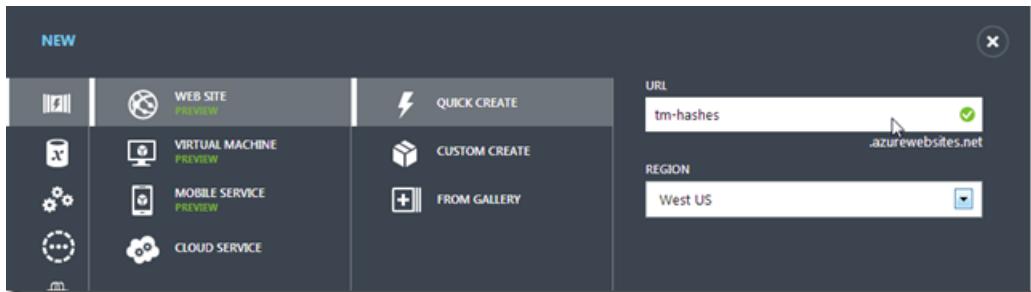
Serge just asked me to create a new TeamMentor (TM) website for him using a particular TM library, so here are the steps I took (note: some of this will be automated in the next TM release)

In Azure

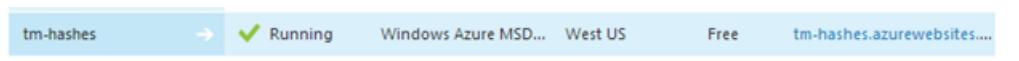
It all started by going into Azure and creating a new website:



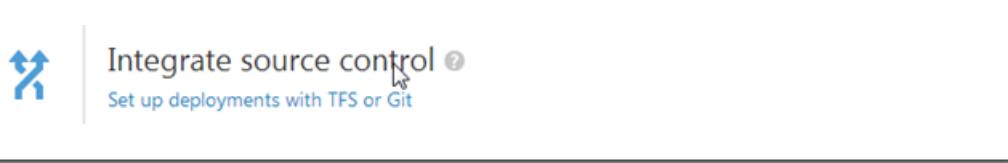
In this case called tm-hashes



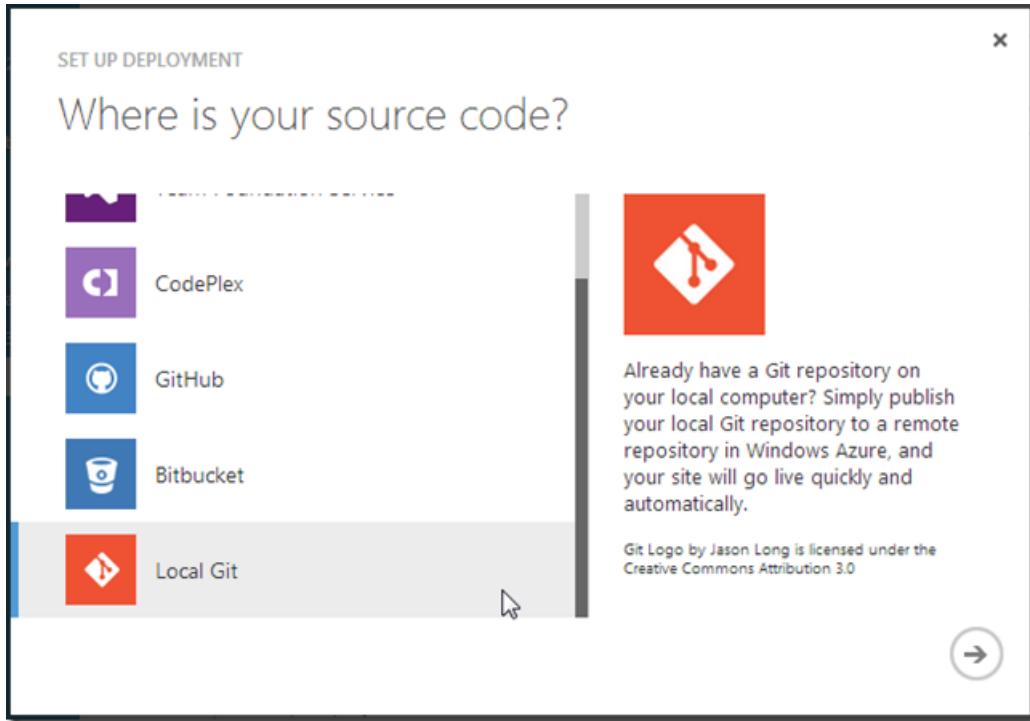
in a couple secs it was available



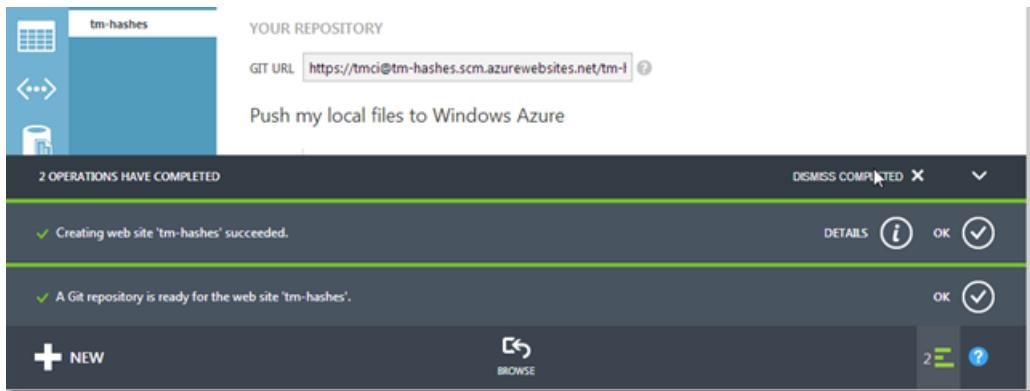
next I set-up git publishing:



using 'Local Git' since that works well with TeamCity and doesn't require that Azure is given pull privileges into the target repo:



Azure worked for a bit, and after a couple secs I had:



In TeamCity ...

Next, in the TeamCity server, to make it easier on next deployments, I added this site to one of the builds that already pushes other sites into Azure:

A screenshot of the TeamCity build configuration interface. The top navigation bar includes 'Projects', 'My Changes', 'Agents', and 'Build Queue'. The main area shows the path 'Administration > TM PowerUsers Project > TM for (Dennis Groves, TM-Vulnerabilities) Configuration'. Below this is a section titled 'General Settings' with three fields: 'Name:' containing 'TM for (Dennis Groves, TM-Vulnerabilities, TM-Hashes)', 'Description:' (empty), and 'Build number format:' with the value '{0}'.

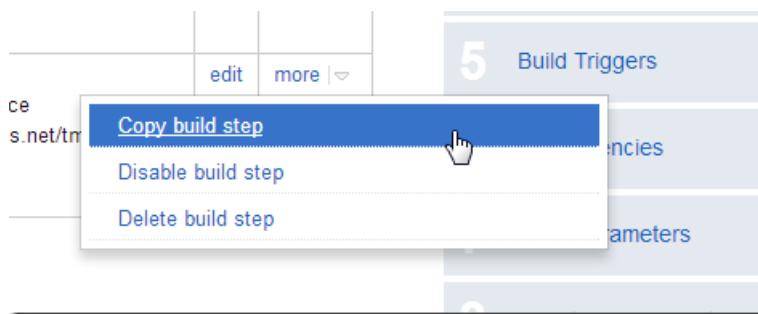
Specifically I added another build-step similar to the ones already there for tm-vulnerabilities and tm-DennisGroves

Build Steps

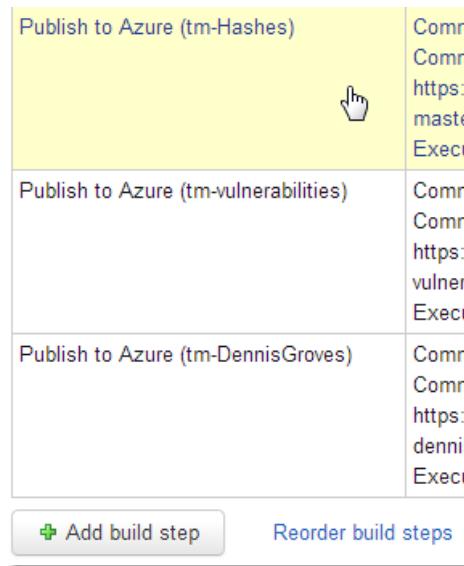
Build Step	Description	edit	more
Visual Studio (.sln)	Build file path: Web Applications/TM_Website.sln Targets: Rebuild Configuration: Release Platform: <default> Execute: Only if all previous steps were successful	edit	more
Copy of Publish to Azure (tm-vulnerabilities)	Command Line Command: C:\Program Files (x86)\Git\bin\git.exe push --force https: [REDACTED]@tm-vulnerabilities.scm.azurewebsites.net/tm-vulnerabilities.git master Execute: Only if all previous steps were successful	edit	more
Publish to Azure (tm-DennisGroves)	Command Line Command: C:\Program Files (x86)\Git\bin\git.exe push --force https: [REDACTED]@tm-dennisgroves.scm.azurewebsites.net/tm-dennisgroves.git master Execute: Only if all previous steps were successful	edit	more

[+ Add build step](#) [Reorder build steps](#)

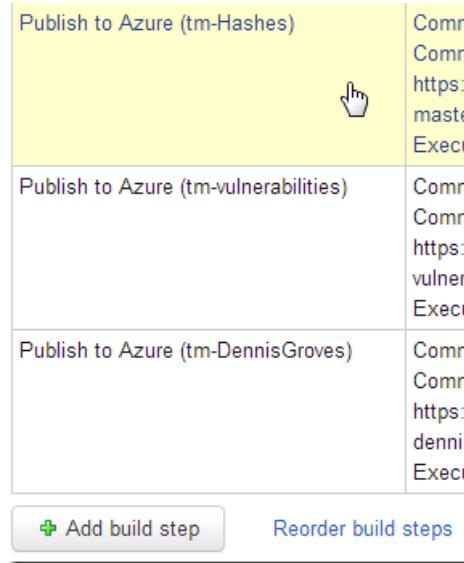
The easiest way to do it, was to create a copy of one of the existing steps:



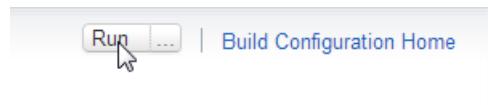
After editing the ‘copied build step’, called *Publish to Azure (tm-Hashes)*, I quickly reordered the build steps from:



to:



Next, I clicked on Run:



in order to trigger a TeamCity build:

Build #	Step	Status	Artifacts	Changes	Duration
#15	ResolveProjectReferences	In Progress	No artifacts	No changes	1m:28s left
#14	Success	Success	No artifacts	dinis.cruz (3)	12 hrs ago (3m:24s)

That build will:

- do a git pull from the latest version of the TM code (currently at 3.3 RC3.01),
- build the VisualStudio main project,
- and push to Azure

Here are the build logs during the ‘push to Azure’ step:

```
[14:30:30] [Web Applications\TeamMentor.UnitTests\TeamMentor.UnitTests.cs]
[14:30:30] [CopyFilesToOutputDirectory] Copy
[14:30:30] [Copy] Copying file from "obj\Release\TeamMentor.UnitTe
[14:30:30] [CopyFilesToOutputDirectory] TeamMentor.UnitTests -> C:\Bui
[14:30:30] [CopyFilesToOutputDirectory] Copy
[14:30:30] [Copy] Copying file from "obj\Release\TeamMentor.UnitTe
[14:30:30] [Step 1/4] Process exited with code 0
[14:30:31] Step 2/4: Publish to Azure (tm-Hashes) (Command Line) (running for 36s)
[14:30:31] [Step 2/4] Starting: "C:\Program Files (x86)\Git\bin\git.exe" push --force https://
[14:30:31] [Step 2/4] in directory: C:\BuildAgent_Service\work\8800230fb9ca9969
```

At this moment the Azure admin page for the tm-hashes site, will show a *Deploying* message

tm-hashes PREVIEW

DASHBOARD DEPLOYMENTS MONITOR CONFIGURE SCALE LINKED RESOURCES

C DEPLOYING: Thursday, March 14, 2013 2:31 PM

N/A

ID: InProgress AUTHOR: N/A DEPLOYED BY: N/A

Last Refreshed: Thursday, March 14, 2013 2:32 PM

which becomes *Active Deployment* once it is completed:

The screenshot shows the Azure portal interface for the 'tm-hashes' application. At the top, there's a navigation bar with 'DASHBOARD', 'DEPLOYMENTS' (which is highlighted in red), 'MONITOR', 'CONFIGURE', 'SCALE', and 'LINKED RESOURCES'. Below the navigation bar, the page title is 'deployment history'. A 'GIT URL' input field contains the value 'https://tmci@tm-hashes.scm.azurewebsites.net/tm-hashes.git'. To the right of the input field is a question mark icon. The main content area displays a deployment log entry. It features a small red square icon with a white 'P' inside. To its right, the text reads: 'ACTIVE DEPLOYMENT: Thursday, March 14, 2013 2:32 PM'. Below this, a message says 'Ignoring (for now) unittest that was failing on RC3'. Underneath the message, deployment details are listed: 'ID: 0758aa6e3c', 'AUTHOR: DinisCruz', and 'DEPLOYED BY: tmci'. At the bottom left of the log area, the text 'Last Refreshed: Thursday, March 14, 2013 2:33 PM' is visible.

One of the nice hacks the Azure team did with their git implementation is to provide good messages/info in the git data sent back on pushes (the lines in dark-orange below where created by Azure):

```
[14:30:31]      [Step 2/4] in directory: C:\BuildAgent_Service\work\8800230fb9ca9969
[14:32:54]      [Step 2/4] remote: New deployment received. [K
[14:32:54]      [Step 2/4] remote: Updating branch 'master'. [K
[14:32:54]      [Step 2/4] remote: ..... [K
[14:32:54]      [Step 2/4] remote: Updating submodules. [K
[14:32:54]      [Step 2/4] remote: Preparing deployment for commit id '0758aa6e3c'. [K
[14:32:54]      [Step 2/4] remote: Building web project 'TM_Website.csproj'. [K
[14:32:54]      [Step 2/4] remote: . [K
[14:32:54]      [Step 2/4] remote:   TeamMentor.CoreLib -> C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site
[14:32:54]      [Step 2/4] remote:   TM_Website -> C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site\reposit
[14:32:54]      [Step 2/4] remote:   Copying all files to temporary location below for package/publish: [K
[14:32:54]      [Step 2/4] remote:   C:\DWASFiles\Sites\tm-hashes\Temp\ee2054b79-ca74-45ea-815b-949b0e53a290. [K
[14:32:54]      [Step 2/4] remote: Preparing files for deployment. [K
[14:32:54]      [Step 2/4] remote: Deployment successful. [K
[14:32:54]      [Step 2/4] To https://[REDACTED]tm-hashes.scm.azurewebsites.net/tm-hashes.git
[14:32:54]      [Step 2/4] * [new branch]    master -> master
[14:32:54]      [Step 2/4] Process exited with code 0
```

Once the push is complete, we can browse the Azure site:



And see an empty TeamMentor installation:

The screenshot shows the TeamMentor control panel at <https://tm-hashes.azurewebsites.net/teamMentor>. The interface includes a sidebar with 'Applied Filters' and 'Guidance Libraries'. A central search area displays a table with columns 'Technology', 'Phase', and 'Type'. A message on the right states 'Login required' and provides fields for 'Username', 'Password', 'Repeat Password', 'E-Mail', and 'First Name'.

In TM Control panel using GitHub zip file ...

The final step is to add the library that Serge wants (note: for this example I'm going to install the library via drag and drop of zip files.)

To get the library files, I went to its private Github repository, and clicked on the **zip** button

The screenshot shows a GitHub repository page for 'TMContent / Lib_Hashes'. The repository is marked as PRIVATE. It features tabs for 'Code', 'Network', 'Pull Requests (0)', 'Issues (0)', and 'Wiki'. Below the tabs, there's a note 'No description or homepage.' and buttons for 'Clone in Windows', 'ZIP' (which is highlighted), 'HTTP', 'SSH', and the repository URL 'https://github.com/TMContent/Lib_Hashes.git'. At the bottom, there are buttons for 'branch: master', 'Files' (which is selected), 'Commits', 'Branches (1)', and a 'T' icon.

which downloaded the zip file into my current vm:

Name	Date modified	Type	Size
Lib_Hashes-master.zip	3/14/2013 2:38 PM	ZIP File	11,647 KB

back in TeamMentor,

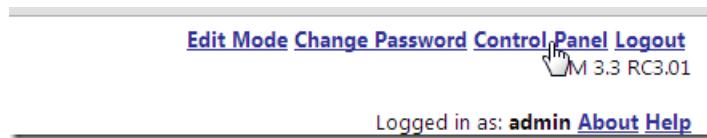
I logged in:



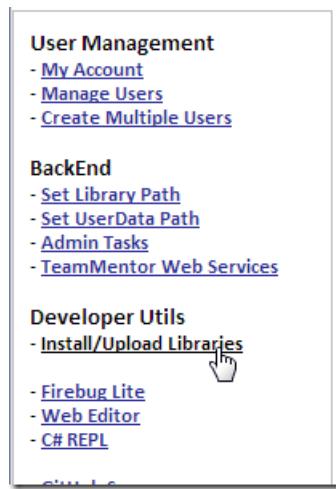
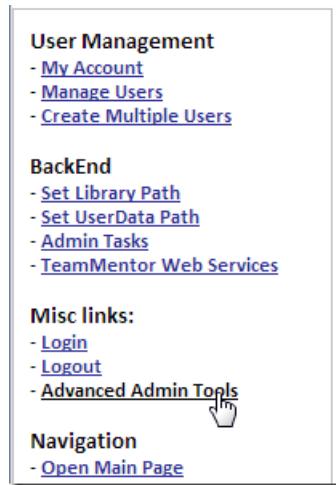
as admin

A screenshot of a modal dialog box titled 'Login into TeamMentor'. It contains two input fields: 'Username' with 'admin' typed in and 'Password' with several dots. Below the fields are two buttons: a blue 'Login' button with a cursor pointing at it, and a link 'Forgot your password?'. The dialog has a close button in the top right corner.

went to the control panel:



clicked on *Advanced Admin Tools* and *Install/Upload Libraries*



drag-n-dropped the local zip file into the *Upload a file* red box

User Management

- [My Account](#)
- [Manage Users](#)
- [Create Multiple Users](#)

BackEnd

- [Set Library Path](#)
- [Set UserData Path](#)
- [Admin Tasks](#)
- [TeamMentor Web Services](#)

Developer Utils

- [Install/Upload Libraries](#)
- [Firebug Lite](#)
- [Web Editor](#)
- [C# REPL](#)
- [GitHub Sync](#)
- [TeamMentor QUnit Tests](#)
- [UIAutomation \(Admin\)](#)
- [UIAutomation \(User\)](#)

Misc links:

- [Login](#)
- [Logout](#)
- [Advanced Admin Tools](#)

Navigation

- [Home](#)
- [Help](#)

Install Library

Library Zip files uploaded to this server (click to install)

Current Libraries Zip Folder(server side): C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase\LibrariesUp...

Note that instalation from Zip will override existing libraries with the same name. [refresh list](#)

Install Library from File or Url

URLs for Libraries to install: [OWASP](#) [Top 20 Vulnerabilities](#) [TM Documentation](#) [Java \(need Java\)](#)

File (server-side) or Url:

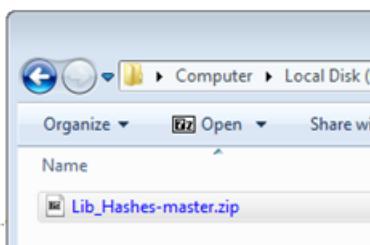
Unzip Password:

Install

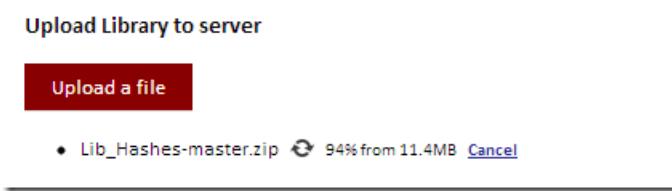
Upload Library to server

Upload a file 

• Lib_Hashes-master.zip 7%from 11.4MB [Cancel](#)



Depending on the network speed, this can take a couple seconds or minutes:



Once the upload was completed,

I clicked on the 'Lib_Hashes-master.zip' link that appeared at the top:

Current Libraries Zip Folder(server side): C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase\LibrariesUploadedFiles

Note that instalation from Zip will override existing libraries with the same name. [refresh list](#)

- [Lib_Hashes-master.zip](#)

Install Library from File or Url

URLs for Libraries to install: [OWASP](#) [Top 20 Vulnerabilities](#) [TM Documentation](#) [Java \(needs unzip pw\)](#)

File (server-side) or Url:

Unzip Password:

Install

Upload Library to server

Upload a file

- Lib_Hashes-master.zip 11.4MB

waited a little bit:

File (server-side) or Url: C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site

Unzip Password:

Install

... installing library: C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase\LibrariesUploadedFiles/Lib_Hashes-master.zip

and once I saw the success message:

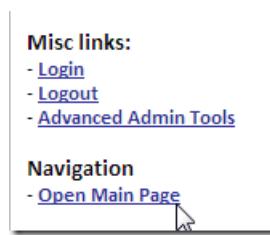
File (server-side) or Url: C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site

Unzip Password:

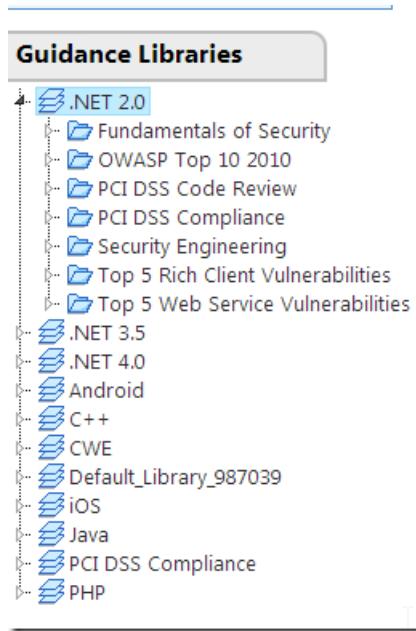
Install

> Library installed was successful

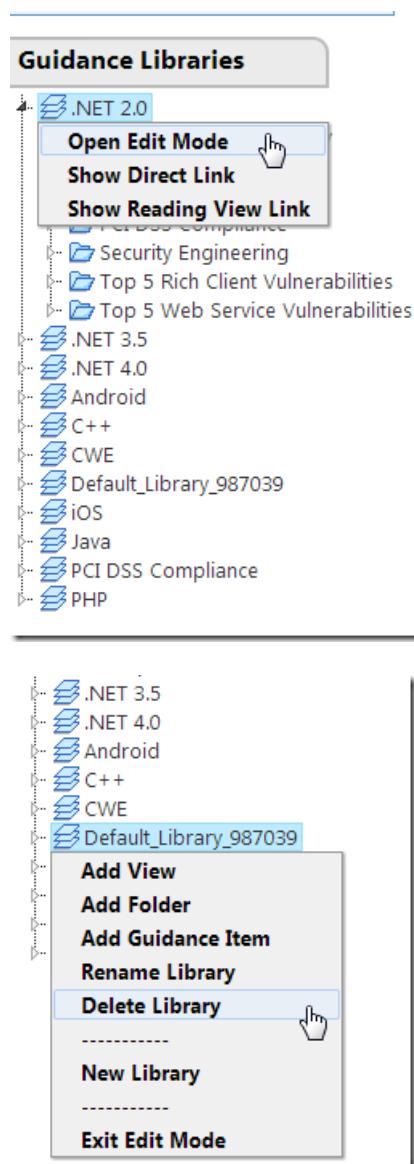
I clicked on **Open Main Page**

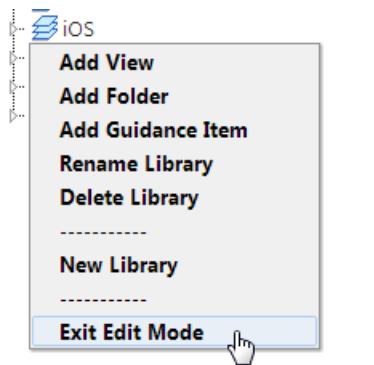


and the main TeamMentor GUI showed the imported libraries:



The final step was to remove the original temp library (created by TeamMentor on first install)





And finally, I sent Serge an email with the link to his brand new install of TeamMentor :)

Title	Technology	Phase	Type
A Centralized Log Server Is Deployed	Any	Deployment	Checklist Item
A Certificate Is Installed on the Database Server to Support SSL Communication	SQL Server 2000	Deployment	Checklist Item
A Control Flow Analysis Is Performed	Any	Implementation	Checklist Item
A Custom ASP.NET Policy Is Used to Access Non-SQL Server Databases from Partial Trust ASP.NET Applications.	ADO.NET 2.0	Implementation	Checklist Item
A Custom Least-privileged Anonymous Account Is Created for Anonymous Access.	Web Application	Deployment	Checklist Item
A Data Flow Analysis Is Performed	Any	Implementation	Checklist Item

Using Git Branches to fix Issues added to TeamMentor's GitHub repository

This is the currently workflow that I'm following when coding/fixing TeamMentor Issues added to the [TeamMentor/Master/Issues¹²](https://github.com/TeamMentor/Master/Issues) list.

- Find issue to address
- Create and checkout new branch (with the issue ID on its title)
- Apply the fixes (on the new branch)
- Commit the changes (on the new branch)
- Checkout master branch
- Merge changes (from new branch) into master , using the -no-ff (no fast-forward) option (this is very important, see [here¹³](#) and [here¹⁴](#) for a good explanation why)
- Push to GitHub

Lets look at this in action.

Here is a simple issue to fix: [https://github.com/TeamMentor/Master/issues/389¹⁵](https://github.com/TeamMentor/Master/issues/389)

tlaloc75 opened this issue 41 minutes ago

Text for signup page should work for SI as well as Customer sites

Assignee: DinisCruz **Milestone:** 3.3 SI Release **Labels:**

Text currently is:
If you have a valid account, please Login. If not, please create an account to have 15 days access.

Should be:
If you have a valid account, please Login. If not, please sign up for an account to gain access.

1 participant

In GitHub, the issue is #389, so on a local clone of the Master repository, we create a branch called **Issue_389** (using the -b switch to create it)

¹²<https://github.com/TeamMentor/Master/Issues>

¹³<http://stackoverflow.com/questions/6701292/git-fast-forward-vs-no-fast-forward-merge>

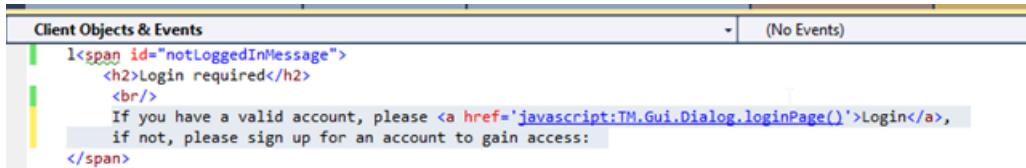
¹⁴<http://stackoverflow.com/questions/2850369/why-does-git-use-fast-forward-merging-by-default/2850413#2850413>

¹⁵<https://github.com/TeamMentor/Master/issues/389>

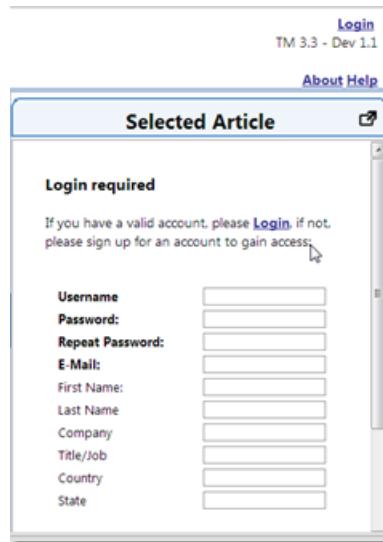
```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git checkout -b Issue_389
Switched to a new branch 'Issue_389'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (Issue_389)
$
```

In VisualStudio, apply the fix to the code:



Quickly look in a browser to confirm the change (this should also be reconfirmed via a UI UnitTest):



Commit the change to the Issue_389 branch:

The screenshot shows the TortoiseGit Pending Changes window for Issue_389. The 'Comments' section contains a note about updating sign-up text and a link to a GitHub issue. The 'Changed files' section lists 'User_Not_Logged.html' under the 'Modified' status. A detailed diff view shows the changes made to the file, including the addition of a login required message and a link to the login page.

```

Pending Changes (Issue_389)
Commit Amend Last Commit Switch Branch Open.gitignore Refresh Git Bash History TortoiseGit
Comments: Sign off
updated sign-up text
Fixes: https://github.com/TeamMentor/Master/issue/389

Changed files (+0 ~1 -0 !0)
Status File
Modified Web Applications/TM_Website/H

```

```

diff --git a/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html b/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html
index 56f1199..7aba32e 100644
--- a/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html
+++ b/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html
@@ -1,7 +1,8 @@
<span id="notloggedinInMessage">
<h2>Login required</h2>
<br/>
- If you have a valid account, please <a href="javascript:TM.Gui.Dialog.loginPage()">Login</a>, if
+ <br/>
+ If you have a valid account, please <a href="javascript:TM.Gui.Dialog.loginPage()">Login</a>,
+ if not, please sign up for an account to gain access:
</span>
<script>

```

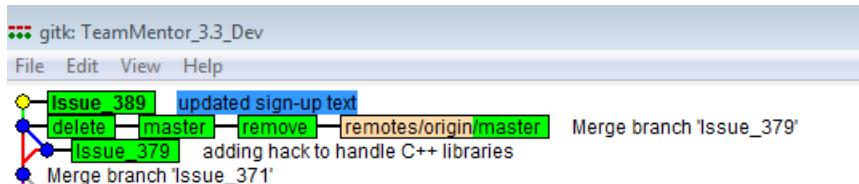
Which means that at this moment, there is nothing else to commit on the *Issue_389* branch:

```

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (Issue_389)
$ git status
# On branch Issue_389
nothing to commit (working directory clean)

```

which is now one commit ahead of master



Next step is to *checkout _ **(into) master and do the **_git merge* using the *_no-ff_*

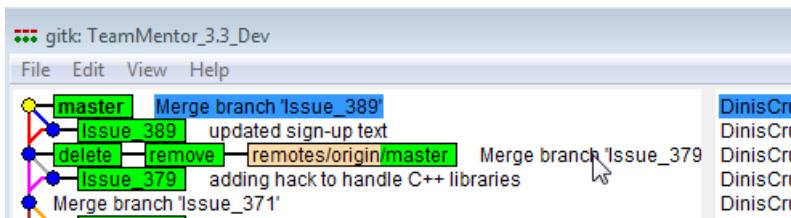
```

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (Issue_389)
$ git checkout master
Switched to branch 'master'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git merge --no-ff Issue_389
Merge made by the 'recursive' strategy.
  .../TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html |      5 +++--
  1 file changed, 3 insertions(+), 2 deletions(-)

```

Gitk shows the effect of the *-no-ff* (ie. the use of the branch was preserved)



Final step is to push the commits to GitHub:

```
G2@WIN-FGNQ5AARJ8O /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git push origin
Counting objects: 16, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 830 bytes, done.
Total 9 (delta 8), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
  4ca6e57..7d488dc master -> master
```

Here is the commit at GitHub:

The screenshot shows a GitHub commit page for a merge commit. The commit message is 'Merge branch 'Issue_389''. It was authored by 'DinisCruz' 6 minutes ago. The commit hash is '7d488dc5db7f20e80f6b484d9aads9a37912e616'. The diff view shows changes to 'User_Not_Logged.html'. The changes include adding a login required message and instructions for both existing and new users.

```
diff --git a/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html b/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html
--- a/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html
+++ b/Web Applications/TM_Website/Html_Pages/Gui/Panels/User_Not_Logged.html
@@ -1,7 +1,8 @@
 1   <span id="notLoggedInMessage">
 2     <h2>Login required</h2>
 3     <br/>
 4     - If you have a valid account, please <a href='javascript:TM.Gui.Dialog.loginPage()">'Login</a>, if not, please cr
 3     +
 4     + If you have a valid account, please <a href='javascript:TM.Gui.Dialog.loginPage()">'Login</a>,
 5     + if not, please sign up for an account to gain access:
 6   </span>
 7
 8   <script>
```

Here is the GitHub's Network view:

The Dev network graph

All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it works.](#)

Show Help



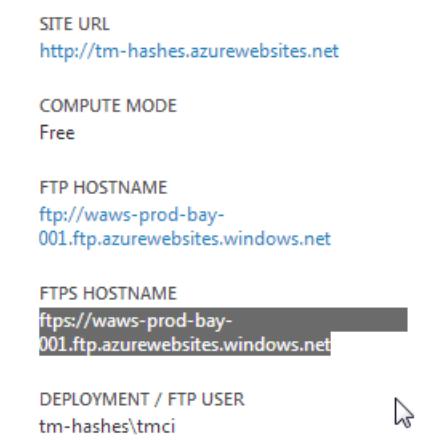
Extracting content files from a Azure deployed version of TeamMentor (pre 3.3 git support)

I was asked by Serge to retrieve some changes he made to a test version of TM hosted in Azure.

This site was hosted at <https://tm-hashes.azurewebsites.net>¹⁶ and since this was a version before the built-in Git Support (where git TM Libraries are natively supported by TM), the only way to get the files was to copy them from the live server.

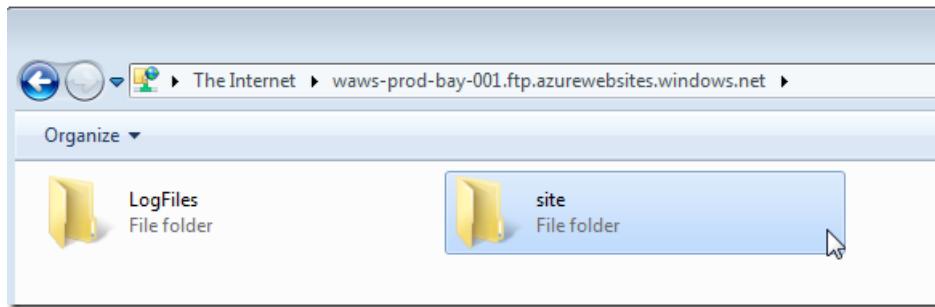
So my first attempt was to use SFTP (which Azure supports) to connect directly to the web root.

To get the SFTP address, I went into Azure's control panel for the tm-hashes site:

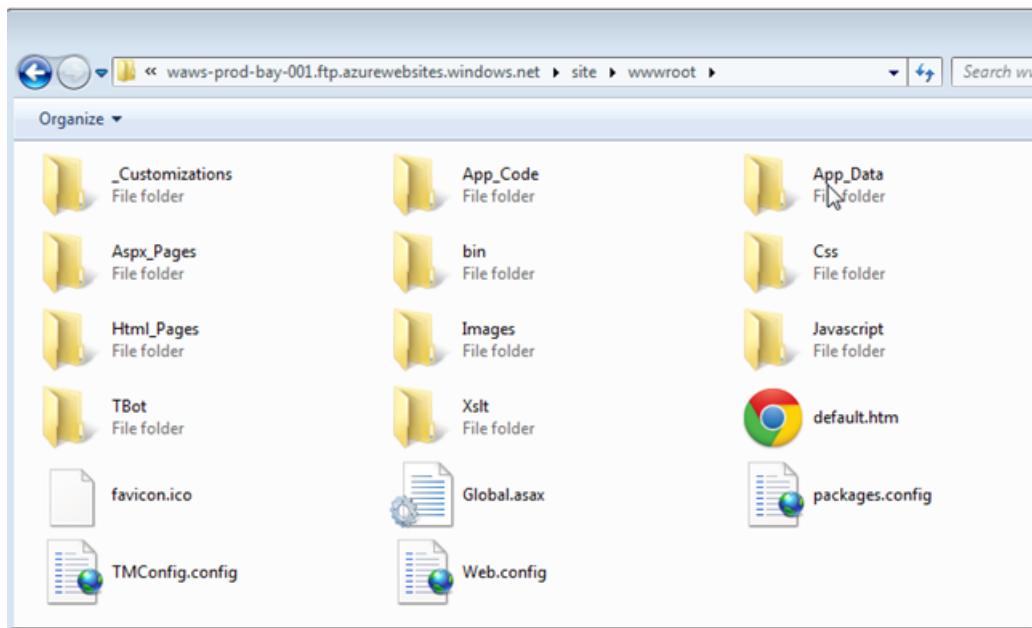


Copied the SFTP address and opened it in local windows explorer (which will require login into the **tm-hashes\tmci** account):

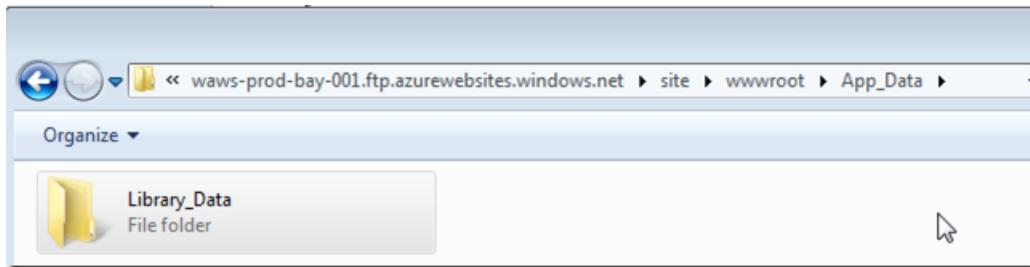
¹⁶<https://tm-hashes.azurewebsites.net/>



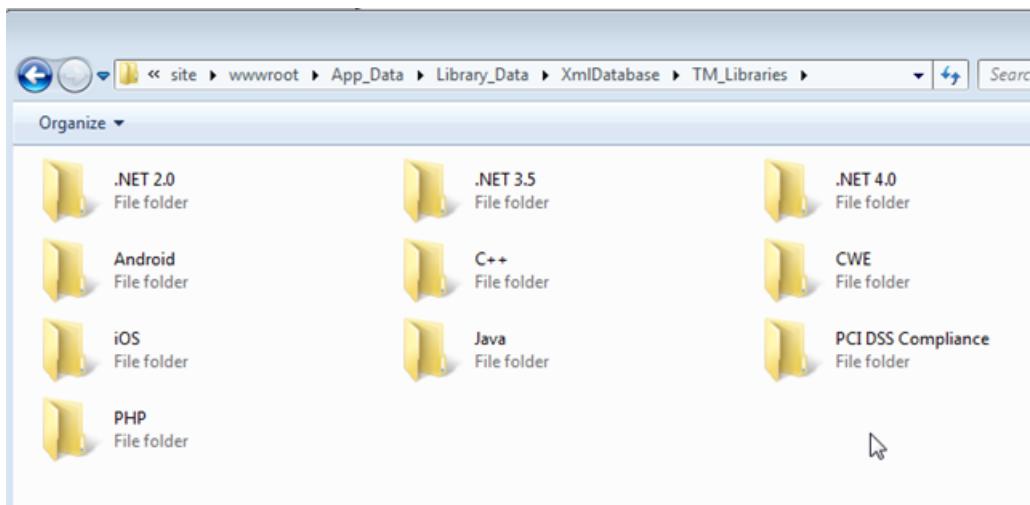
Navigated to the site's webroot (which are the deployed files)



Then into TM's **Library_Data** folder, which is located inside the **App_Data** folder (because this is the main location where this site's IIS account has 'write privileges')

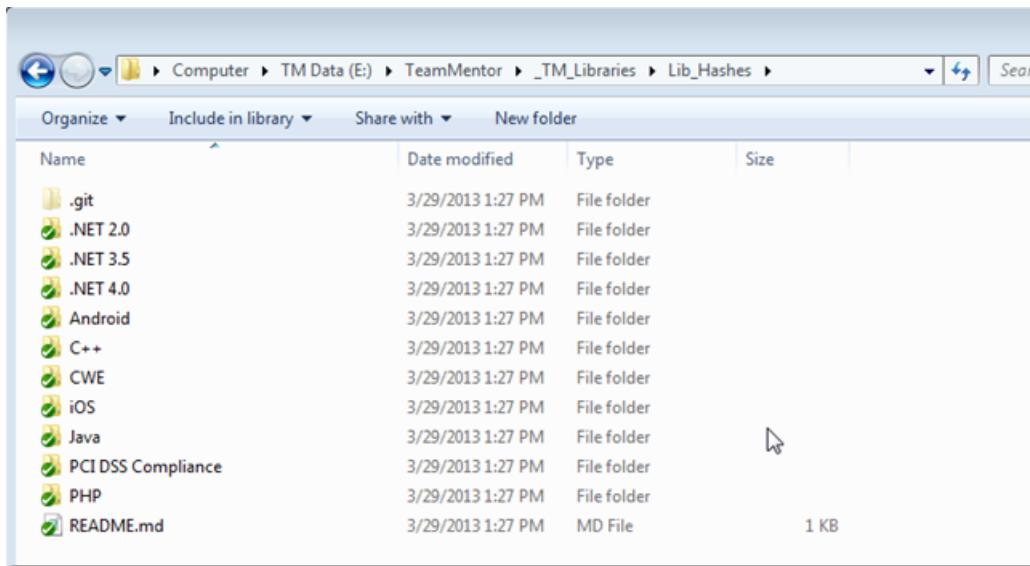


In there we can find the **XmlDatabaseTM_Libraries** folder which contains all libraries currently loaded in this server

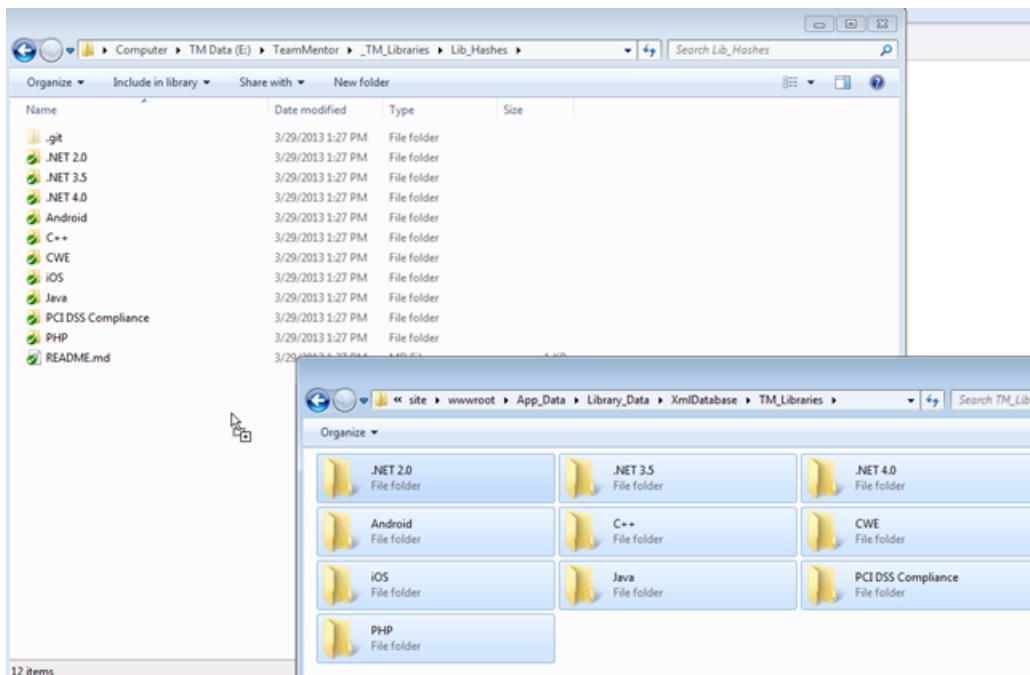


Back in my local VM, I have clone of the target git repository (https://github.com/TMContent/Lib_Hashes¹⁷):

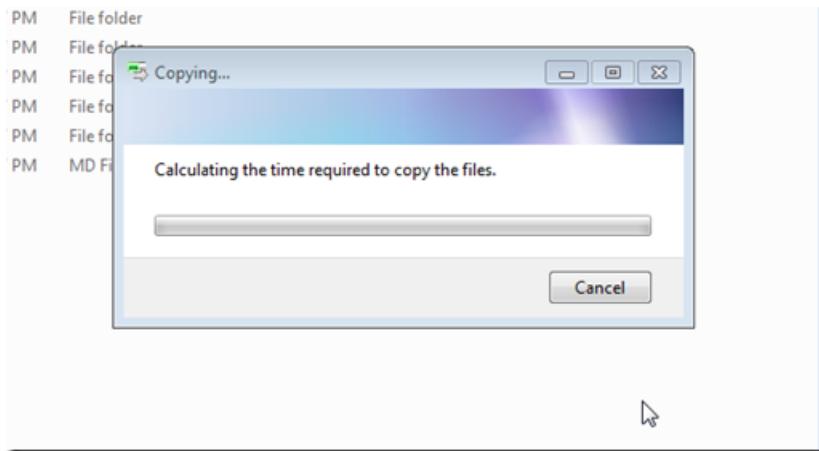
¹⁷https://github.com/TMContent/Lib_Hashes



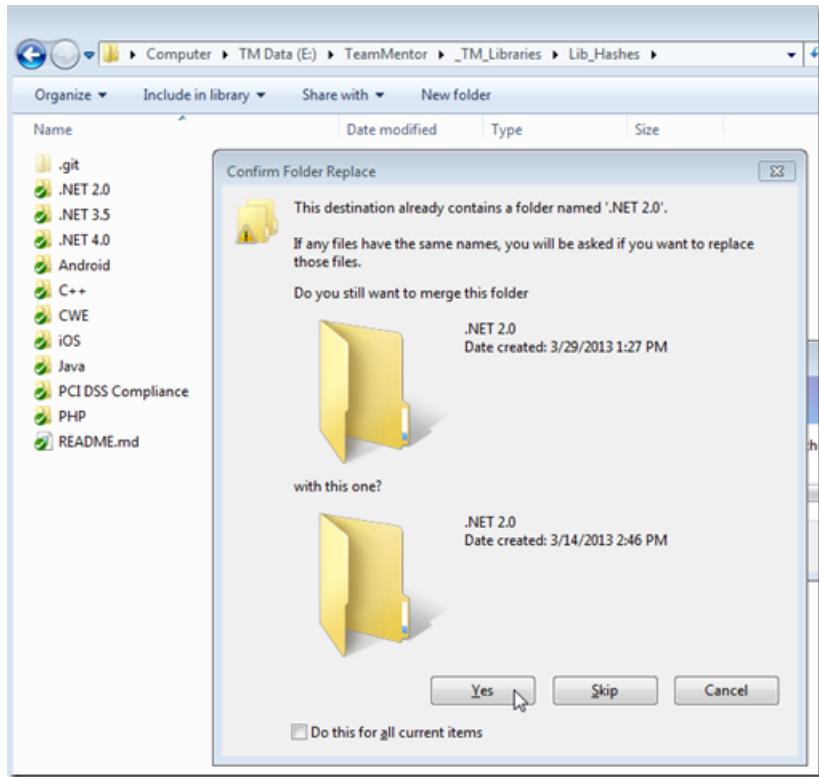
into which I'm going to copy the folders from the SFTP live site:



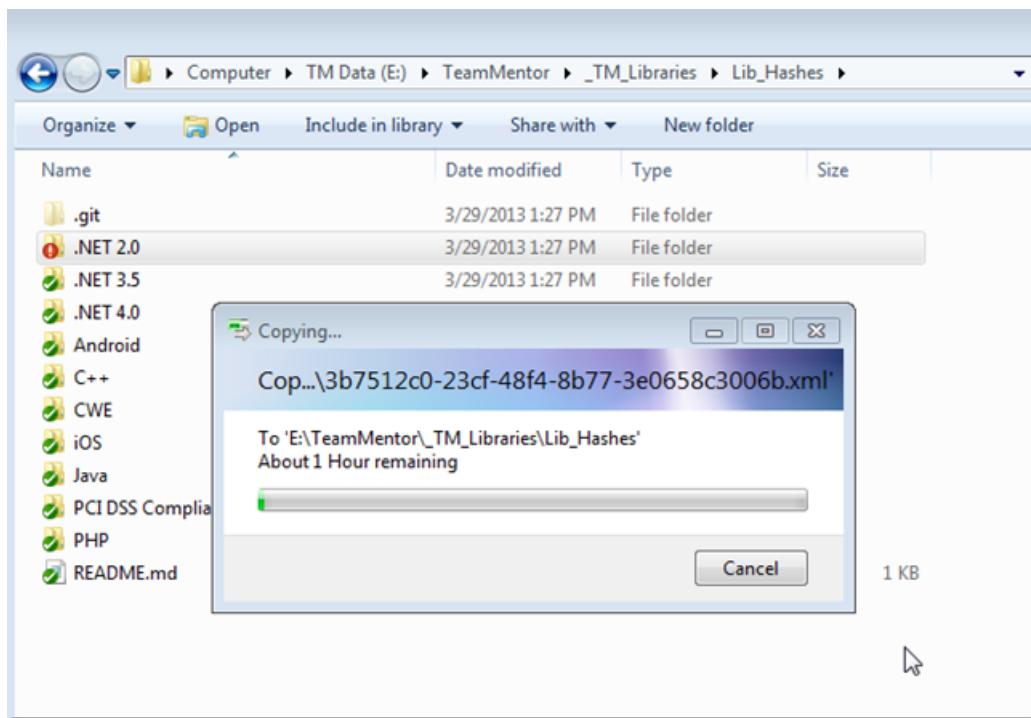
After some time of starting the drop



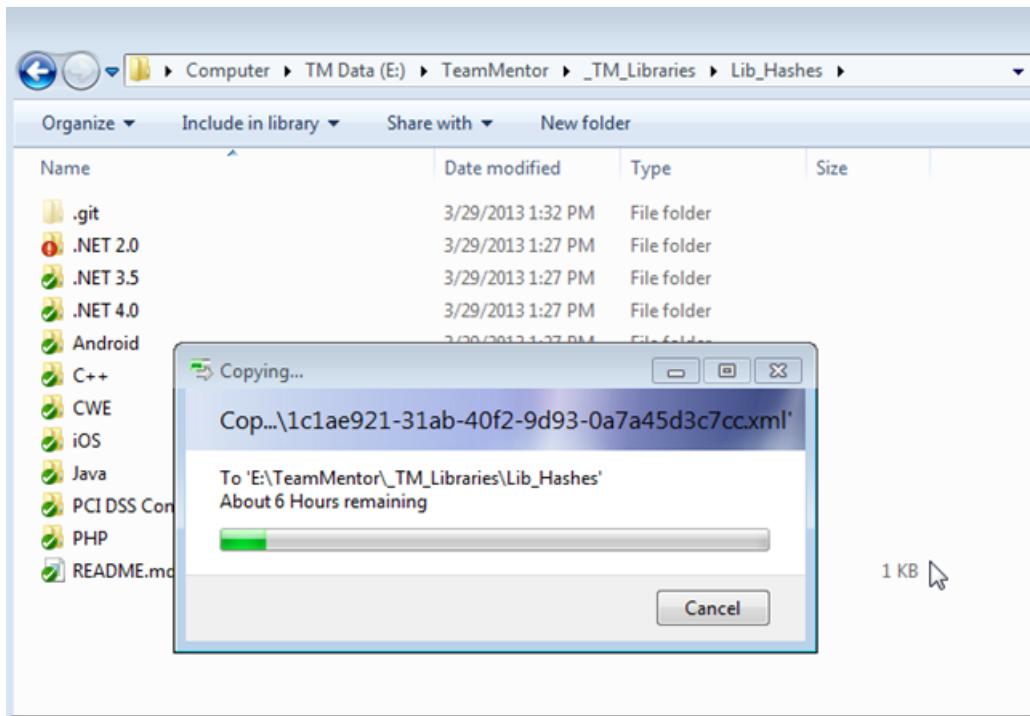
.... I confirm the overwrite



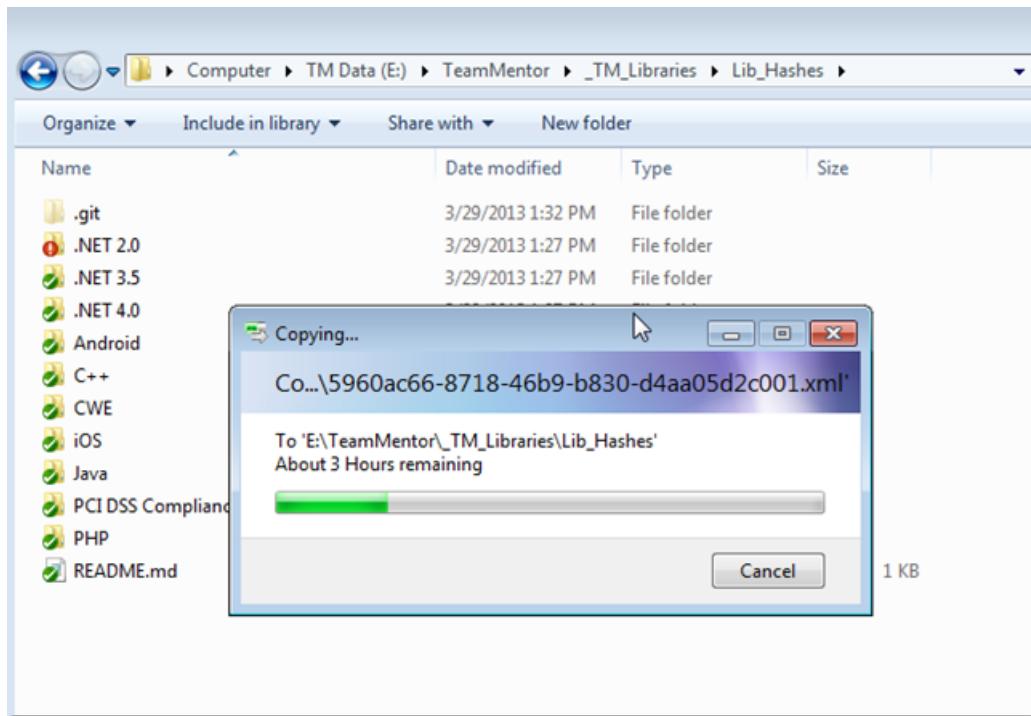
.... and it looks like it will take 1 hour



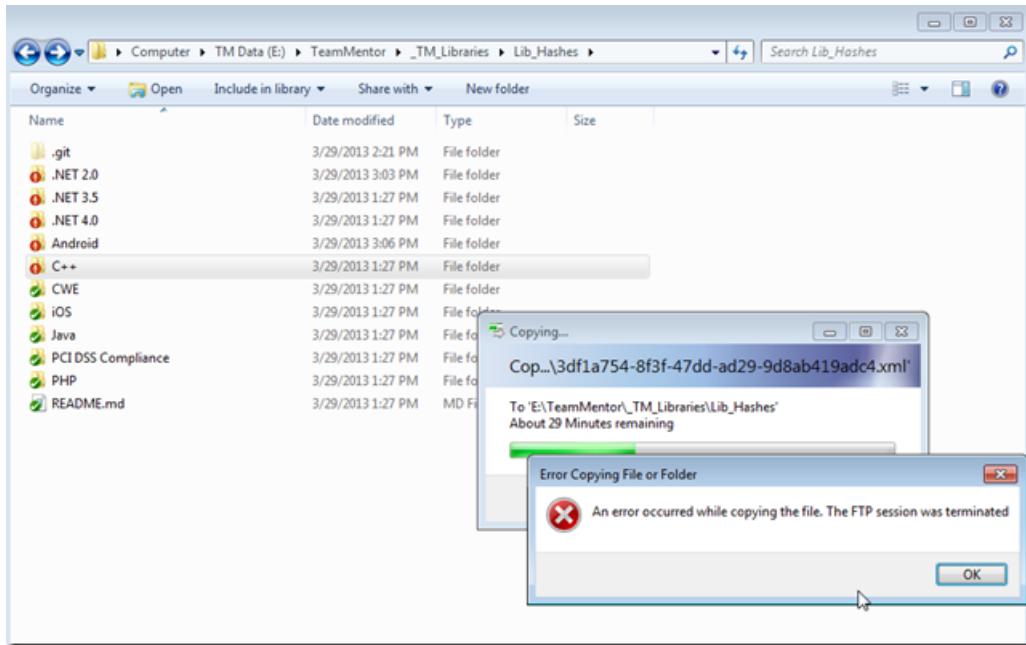
or maybe 6 hours :



After 38m of copying it is now down to 3h



hummm.... damm (after 1h and a bit of copying)



That kinda sucks

I think I need to try a different approach :)

Time to open up the C# REPL included in TM:

```

1 return TM_Xml_Database.Current.Path_XmlDatabase;
2 //using TeamMentor.CoreLib;
3 //O2Ref:TeamMentor.CoreLib.dll

```

Execute

C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site\wwwroot\app_Data\Library_Data\xmlDatabase

Get the path to the Xml Libraries folder:

Web C# REPL is a simple wrapper on O2 Platform's FluentSharp.CoreLib API
Note that all your script executions are logged (see link on top) so don't put any sensitive data here :)

```
1 return TM_Xml_Database.Current.Path_XmlLibraries;
2 //using TeamMentor.CoreLib;
3 //O2Ref:TeamMentor.CoreLib.dll
```

Execute 

C:\DWASFiles\Sites\lm-hashes\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase\TM_Libraries

Confirm that the folders we want are in there:

Web C# REPL is a simple wrapper on O2 Platform's FluentSharp.CoreLib API
Note that all your script executions are logged (see link on top) so don't put any sensitive data here :)

```
1 var librariesPath = TM_Xml_Database.Current.Path_XmlLibraries;
2 return librariesPath.dirs().join("<br/>").remove(librariesPath);
3 //using TeamMentor.CoreLib;
4 //O2Ref:TeamMentor.CoreLib.dll
```

Execute 

\.NET 2.0
\.NET 3.5
\.NET 4.0
\Android
\C++
\CWE
\iOS
\Java
\PCI DSS Compliance
\PHP

Lets first try to zip the forth library:

Web C# REPL is a simple wrapper on O2 Platform's FluentSharp.CoreLib API
Note that all your script executions are logged (see link on top) so don't put any sensitive data here :)

```
1 var librariesPath = TM_Xml_Database.Current.Path_XmlLibraries;
2 var targetFolder = librariesPath.dirs().fourth();
3 return targetFolder;
4 //using TeamMentor.CoreLib;
5 //O2Ref:TeamMentor.CoreLib.dll
```

[Execute](#)

C:\DWASFiles\Sites\lm-hashes\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase\TM_Libraries\Android

Which is the Android one:

Web C# REPL is a simple wrapper on O2 Platform's FluentSharp.CoreLib API
Note that all your script executions are logged (see link on top) so don't put any sensitive data here :)

```
1 var librariesPath = TM_Xml_Database.Current.Path_XmlLibraries;
2 var targetFolder = librariesPath.dirs().fourth(); // Android Library
3
4 var zipFileName = "{0}.zip".format(targetFolder.folderName());
5 var targetZipFile = librariesPath.pathCombine(zipFileName);
6
7 return targetFolder.zip_Folder(targetZipFile); // returns path to zip file
8
9 //using TeamMentor.CoreLib;
10 //O2Ref:TeamMentor.CoreLib.dll
11 //O2Ref:FluentSharp.REPL.exe
```

[Execute](#)

C:\DWASFiles\Sites\lm-hashes\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase\TM_Libraries\Android.zip

After the script shown above executes, a quick look at the ftp site shows the expected Android.zip in there:

The screenshot shows a Windows File Explorer window with the following details:

Address bar: site > wwwroot > App_Data > Library_Data > XmlDatabase > TM_Libraries

Search bar: Search TM_Libr

Organize dropdown menu

Table headers: Name, Size, Type, Date modified, Date created

Data rows:

Name	Size	Type	Date modified	Date created
.NET 2.0		File folder	3/14/2013 2:46 PM	3/14/2013 2:46 PM
.NET 3.5		File folder	3/14/2013 2:46 PM	3/14/2013 2:46 PM
.NET 4.0		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
Android		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
C++		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
CWE		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
iOS		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
Java		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
PCI DSS Compliance		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
PHP		File folder	3/14/2013 2:47 PM	3/14/2013 2:47 PM
Android.zip	62 KB	ZIP File	3/29/2013 6:16 PM	3/29/2013 6:16 PM

Which contains the files we want to get:

The screenshot shows the WinRAR interface displaying the contents of the 'Android.zip' file:

File path: C:\Users\o2\Downloads\Android[1].zip\

Toolbar buttons: Add, Extract, Test, Copy, Move, Delete, Info

File list table:

Name	Size	Packed Size	Modified	Created	Accessed	Attr
_GuidanceItems	132 085	50 047	2013-03-14 14:47	2013-03-14 14:47	2013-03-14 14:47	
README.md	23	16	2013-03-14 14:46	2013-03-14 14:47	2013-03-14 14:47	
Android.xml	2 708	561	2013-03-14 14:47	2013-03-14 14:47	2013-03-14 14:47	

Status bar: 0 object(s) selected

Next lets do the zip of the entire Libraries folder:

Web C# REPL is a simple wrapper on O2 Platform's FluentSharp.CoreLib API
Note that all your script executions are logged (see link on top) so don't put any sensitive data here :)

```
1 var librariesPath = TM_Xml_Database.Current.Path_XmlLibraries;
2
3 var zipFileName  = "{0}.zip".format(librariesPath.folderName());
4 var targetZipFile = librariesPath.parentFolder().pathCombine(zipFileName);
5
6 return librariesPath.zip_Folder(targetZipFile);      // zip the entire librariesPath folder
7
8 //using TeamMentor.CoreLib;
9 //O2Ref:TeamMentor.CoreLib.dll
10 //O2Ref:FluentSharp.REPL.exe
```

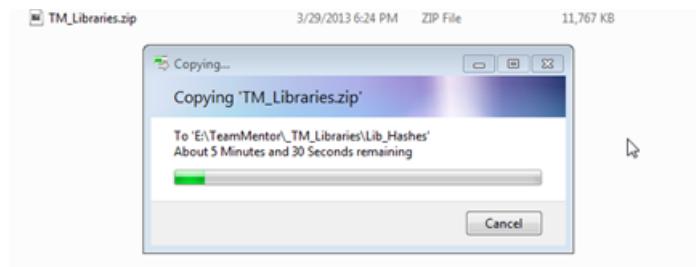
[Execute](#)

C:\DWASFiles\Sites\tm-hashes\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\Xml\Database\TM_Libraries.zip

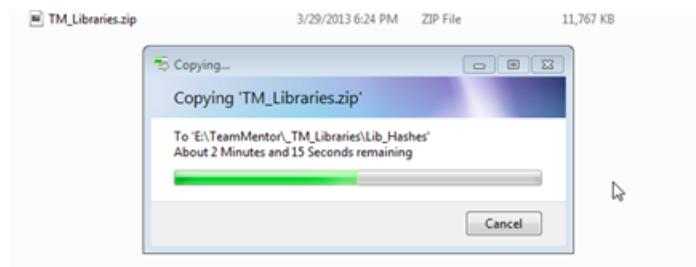
It takes about 1m to create the 11Mb file

Name	Size	Type	Date modified	Date created
Application_Logs		File folder	3/29/2013 5:39 PM	3/29/2013 5:39 PM
LibrariesUploadedFiles		File folder	3/14/2013 2:45 PM	3/14/2013 2:45 PM
TM_Libraries		File folder	3/29/2013 6:18 PM	3/29/2013 6:18 PM
User_Data		File folder	3/26/2013 3:15 AM	3/26/2013 3:15 AM
Cache_guidanceItems.cacheXml	27,196 KB	CACHEXML File	3/14/2013 4:13 PM	3/14/2013 4:13 PM
TM_Libraries.zip	11,767 KB	ZIP File	3/29/2013 6:22 PM	3/29/2013 6:22 PM

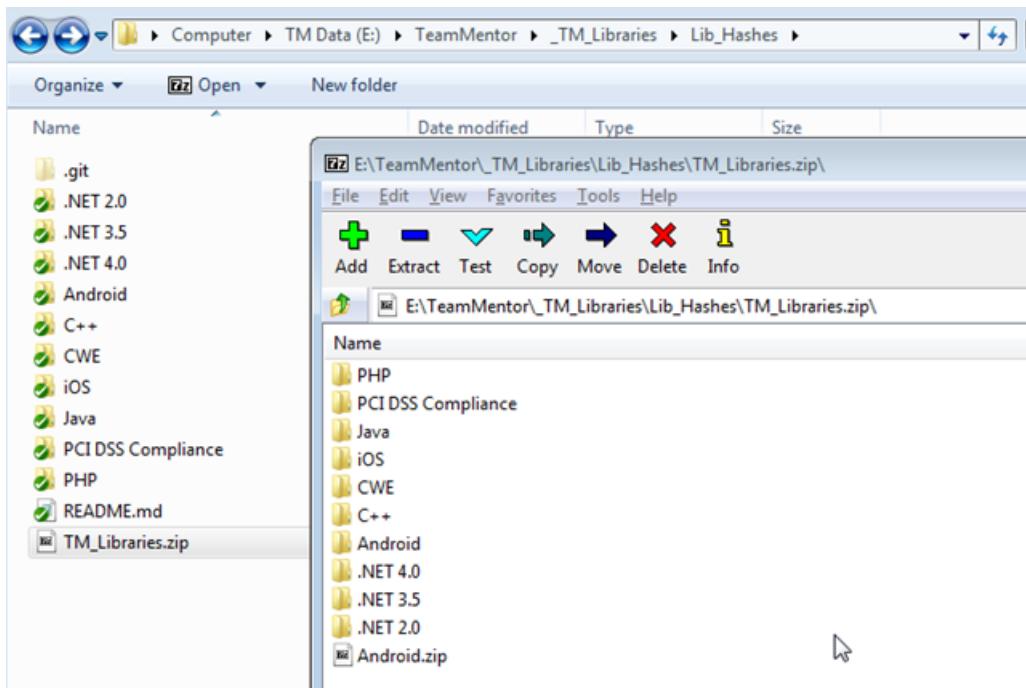
Which can now be copied locally



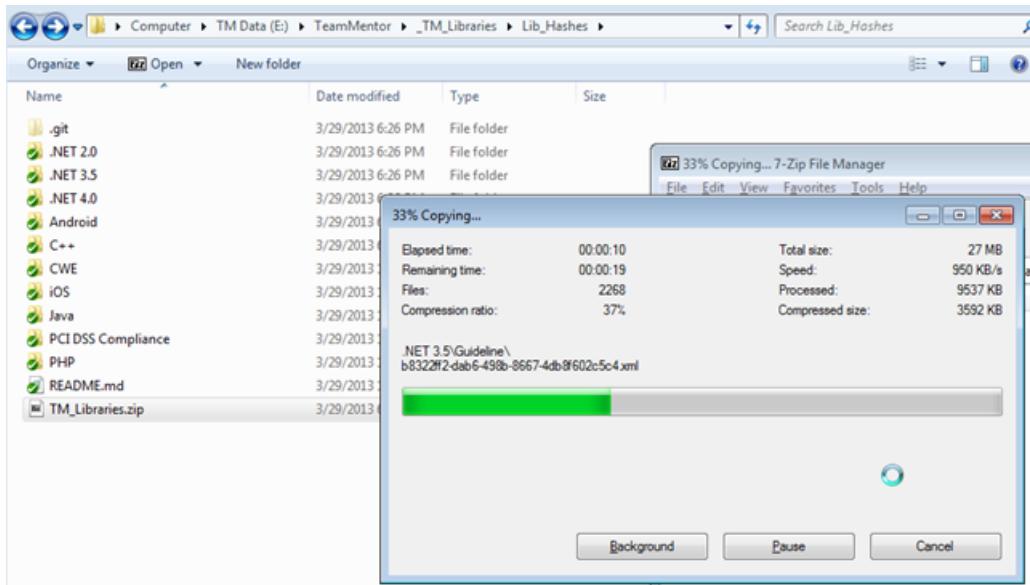
...note how this is much faster than the multiple hours wait we experienced above:



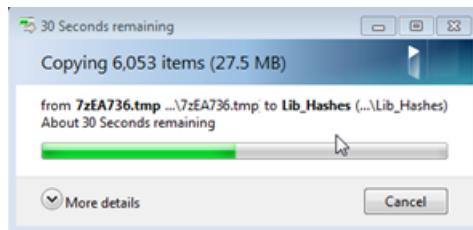
Once the zip file is downloaded:



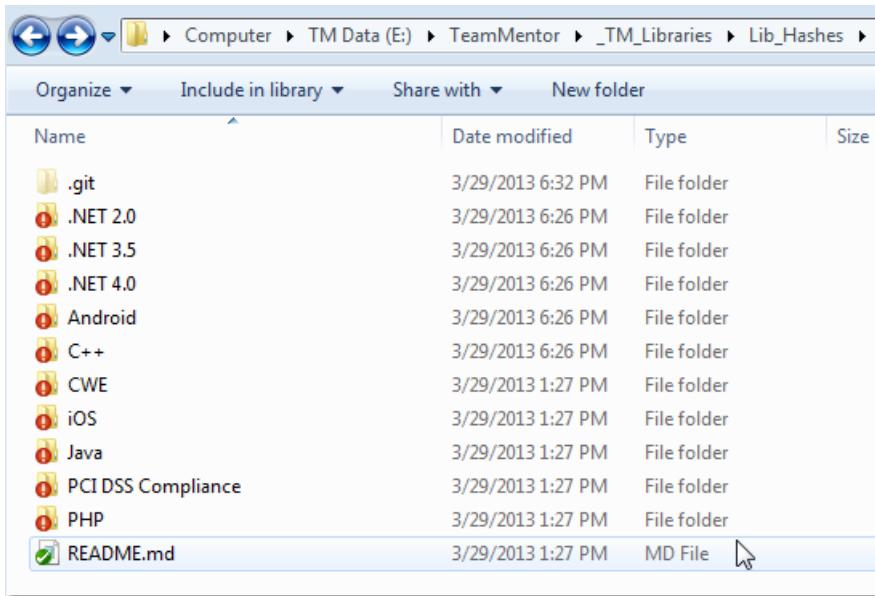
I unzipped the files into the Lib_Hashes folder:



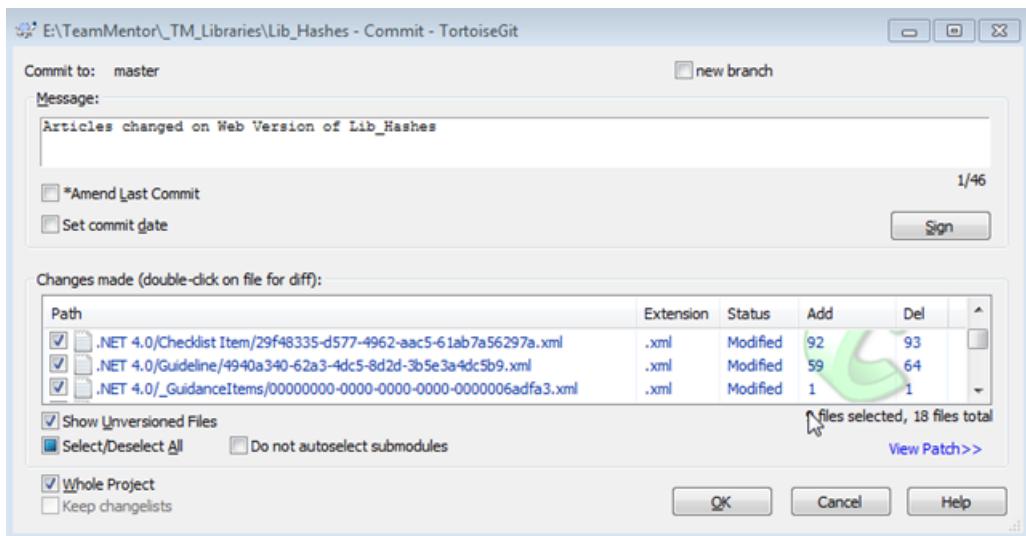
Choosing to overwrite the existing files:



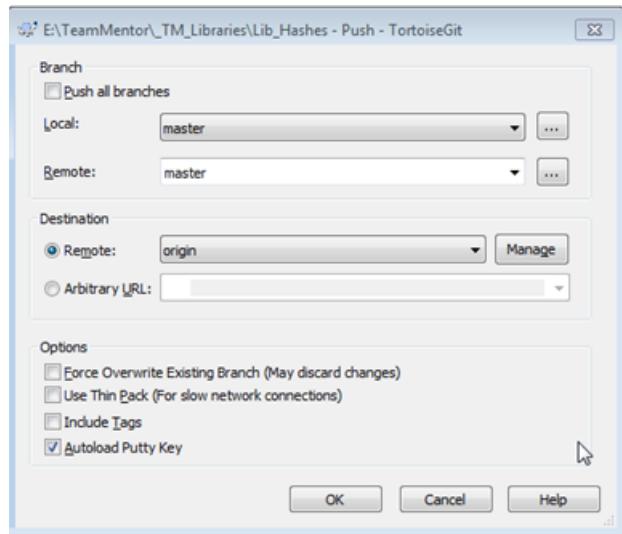
After the copy, Git will pick up the changed files:



Which we can commit:



And push to GitHub:



Using NGit to create native Git support in Azure deployed app (with automatic pushes and pulls)

This entry will show a pretty powerful new feature in [TeamMentor¹⁸](#) (TM) which I'm very proud and excited about!

This feature is so important, that it literary caused a delay on the release of TM 3.3 for about 1 month (my instinct was pushing me on this direction since I 'knew' that this could be done, and that it would be a killer feature). Btw, there is a lot more NGit/Git support than what is shown here, but I'm sure you will see the power in the workflow described below.

Basically, TM's backend engine will now automatically perform:

- a **git pull** when the TM server starts (or it cache is rebuilt)
- a **git commit** followed by a **git push** on every library edit (on both content and structure changes).

And since TM uses the .NET Library [NGit¹⁹](#), what we have here is a pretty powerful self-contained .Net-based 'git for content versioning' solution.

Practically speaking, **this is a Git workflow that runs on Azure-hosted-site without requiring Git to be installed on the live servers!**

This solves the problem created by the lack of git.exe (and supporting files) on an Azure's deployed web application (Azure's git support is limited to pushing code to Azure's servers, which will trigger MSBuild-like website publishing workflow)

Here is a example of this managed Git workflow in action.

The [---

¹⁸<http://teammentor.net/>](https://tm-tm4tm.azurewebsites.net²⁰ site:</p></div><div data-bbox=)

¹⁹<http://blog.diniscruz.com/search/label/NGit>

²⁰<https://tm-tm4tm.azurewebsites.net/>

The screenshot shows the TeamMentor application interface. At the top, there's a header with the TeamMentor logo, a search bar, and links for 'Edit Mode', 'Change Password', 'Logout', 'TM 3.3 - Dev 1.1', and 'Logged in as: DinisCruz About Help'. On the left, there's a sidebar titled 'Applied Filters' and 'Guidance Libraries' containing a tree view of categories like TM4TM, Content Writing Process, Development Process, Get Involved, QA, Guidance Articles, Draft Content, and TeamMentor Architecture. The main area has two filter panels: 'Technology' (listing .Net, Browser, SSL, TeamMentor) and 'Phase' (listing Article Creation, Article Develop..., Contact, Development, Implementation, QA). Below these is a table titled 'Showing 41 items (out of 41)' with columns for Title, Technology, Phase, and Type. The table lists several items such as 'Additional Resources', 'Article Pair Example', 'Attack', 'Checklist Item', 'Checklist Item Format', and 'Client Side ...'. On the right, there's a 'Selected Article' panel and an 'Additional Resources' panel which contains links to the current version of TeamMentor content, a public OWASP TeamMentor library, TeamMentor UI documentation, and TeamMentor schemas.

is currently configured to use the UserData from this repository

The screenshot shows a GitHub repository page for 'Site_tm4tm.teammendor.net'. The repository has 435 commits. The 'Code' tab is selected. The page includes a clone button for Windows, ZIP, HTTP, and SSH, and a link to the repository's URL. It also shows branches (master), files, commits, and branches. The commit history shows changes to 'userData.xml' and other configuration files like 'TMConfig.config' and 'TMSecretData.config'.

Author	Date	Commit Message
unknown-user	3 days ago	changed: Users/admin_e391b43c-7c69-4de5-ba3a-10ea06007e80.userData.xml
Users	just now	changed: Users/admin_e391b43c-7c69-4de5-ba3a-10ea06007e80.userData.xml [unknown-user]
TMConfig.config	8 days ago	Update TMConfig.config [SergeTruth]
TMSecretData.config	9 days ago	Setting SMTP password on TMSecretData.config [DinisCruz]

which contains a reference to the https://github.com/TMContent/Lib_TM4TM²¹ repository:

²¹https://github.com/TMContent/Lib_TM4TM

The screenshot shows the GitHub repository page for `TMContent / Lib_TM4TM`. The repository has 12 issues and 0 pull requests. The commit history shows 78 commits from an unknown user, with the latest commit being `879396cc3c`. The commits are listed in descending order of age, with the most recent at the top. The commits include changes to various files such as `Articles/0b9c58ef-0048-4d02-b98d-51b2bea11ef1.xml`, `Content Writing Process/_GuidanceItems/00000000-0000-0000-0000-000000000000`, and `Development Process/_Images/Contact.png`.

File	Age	Commit Message
<code>Articles</code>	2 days ago	changed: Articles/0b9c58ef-0048-4d02-b98d-51b2bea11ef1.xml [unknown-user]
<code>Content Writing Process</code>	9 days ago	changed: Content Writing Process/_GuidanceItems/00000000-0000-0000-0000-000000000000 [unknown-user]
<code>Development Process</code>	9 days ago	Added: _Images/Contact.png , _Images/Guidance Relationships.jpg , _Im... [DinisCruz]
<code>Guidance Articles</code>	9 days ago	removed: Guidance Articles/_GuidanceItems/00000000-0000-0000-0000-000000000000 [unknown-user]
<code>QA</code>	4 months ago	Added library files. [Roman87]
<code>_Images</code>	9 days ago	Added: _Images/Contact.png , _Images/Guidance Relationships.jpg , _Im... [DinisCruz]

In practice, what this means is that the TM articles we see in <https://tm-tm4tm.azurewebsites.net>²² are the ones hosted and managed by the https://github.com/TMContent/Lib_TM4TM²³ repository.

But since the mapping is done via NGit and the account used (in Azure) to connect to GitHub's Lib_TM4TM has push privileges, it is now possible to make a change in <https://tm-tm4tm.azurewebsites.net>²⁴ that is auto committed locally and then into https://github.com/TMContent/Lib_TM4TM²⁵

Auto Committing and pushing changes

For example, here are the last commits at https://github.com/TMContent/Lib_TM4TM²⁶

²²<https://tm-tm4tm.azurewebsites.net/>

²³https://github.com/TMContent/Lib_TM4TM

²⁴<https://tm-tm4tm.azurewebsites.net/>

²⁵https://github.com/TMContent/Lib_TM4TM

²⁶https://github.com/TMContent/Lib_TM4TM

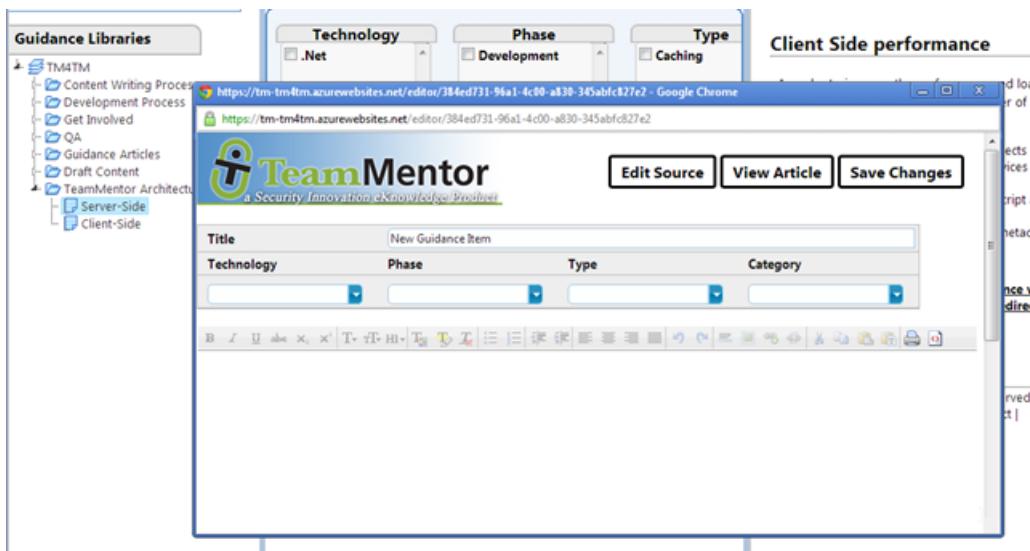
The screenshot shows a GitHub repository page for 'Lib_TM4TM'. At the top, there are tabs for 'Code', 'Network', 'Pull Requests', 'Issues', 'Wiki', 'Graphs', and 'Settings'. Below these are dropdowns for 'branch: master', 'Files', 'Commits' (which is selected), 'Branches' (with a count of 1), and 'Tags'. A 'Keyboard shortcuts available' link is at the top right. The main area shows a commit history for March 28, 2013. Two commits are listed, both authored by an unknown user 2 days ago. The first commit, f7cc17fd2d, is highlighted with a mouse cursor.

In the [https://tm-tm4tm.azurewebsites.net²⁷](https://tm-tm4tm.azurewebsites.net) server, let's add a new Guidance Item (i.e. an Article)

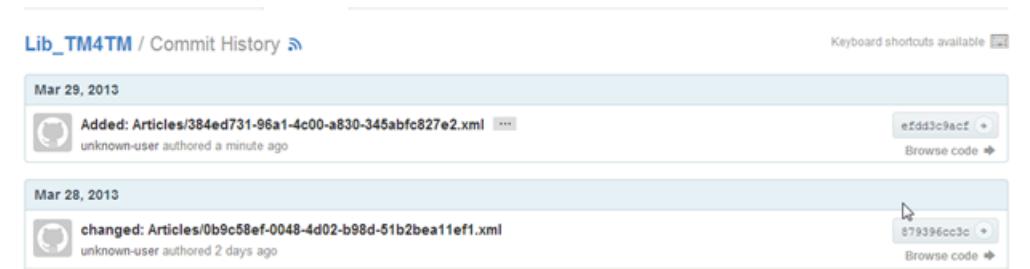
The screenshot shows the 'TeamMentor' web application. The URL is https://tm-tm4tm.azurewebsites.net/teamMentor. The interface includes a 'Search' bar and three filter panels: 'Technology' (with '.Net'), 'Phase' (with 'Development'), and 'Type' (with 'Caching'). Below these filters, a message says 'Showing 1 items (out of 1)'. A table lists one item: 'Client Side performance' under 'Index', with columns for Title, Technology (.Net), Phase (Development), Type (Caching), and Category (TM Architecture). On the left, a sidebar titled 'Applied Filters' shows 'Guidance Libraries' with categories like TM4TM, Content Writing Process, Development Process, Get Involved, QA, Guidance Articles, Draft Content, and TeamMentor Architecture. Under 'TeamMentor Architecture', 'Server-Side' is selected. A context menu is open over this category, showing options: 'Add Guidance Item' (which is highlighted with a cursor), 'Rename View', 'Delete View', a separator line, and 'Exit Edit Mode'.

Since (in TM 3.3) the articles are created immediately, by the time the editor is shown, the article shown in the popup-window will already exist on disks (i.e. there is already an **384ed731-96a1-4c00-a830-345abfc827e2.xml** file on the server)

²⁷<https://tm-tm4tm.azurewebsites.net/>



And with the new ‘auto Git commit’ feature, the git commit of the new article will be available (after a couple secs) at GitHub:



This ‘commit on new article’ is made of two file changes:

- The new article (the `384ed731-96a1-4c00-a830-345abfc827e2.xml` file)
- The mapping of the new Article’s GUID to the chosen ‘view’ element (which is part of the `Tm4TM.xml` library xml file)

Added: Articles/384ed731-96a1-4c00-a830-345abfc827e2.xml

changed: TM4TM.xml

unknown-user authored 8 minutes ago 1 parent 879396c commit efdd3c9acf2c177249cc99f1ff4c12552e3e6c01

Showing 2 changed files with 18 additions and 0 deletions. Show Diff Stats

View file @ efdd3c9

17 Articles/384ed731-96a1-4c00-a830-345abfc827e2.xml

```
@@ -0,0 +1,17 @@
1 <?xml version="1.0"?>
2 <TeamMentor_Article Metadata_Hash="0" Content_Hash="0">
3   + <Metadata>
4     + <Id>384ed731-96a1-4c00-a830-345abfc827e2</Id>
5     + <Library_Id>13e0ebf0-2864-4031-9be7-3de4537ed3a9</Library_Id>
6     + <Title>New Guidance Item</Title>
7     + <Category />
8     + <Technology />
9     + <Type />
10    + <Author />
11    + <Priority />
12    + <Status />
13   + </Metadata>
14   + <Content Sanitized="false" DataType="html">
15     + <Data><![CDATA[]]></Data>
16   + </Content>
17 </TeamMentor_Article>
```

1 TM4TM.xml

```
@@ -133,6 +133,7 @@
133   <view id="32674083-1e8c-4135-8621-57467400761a" caption="Server-Side" creationDate="2013-03-26T13:17:10.809
134     <items>
135       <item>8bb287be-518d-4ed1-9063-07c277aad665</item>
136     + <item>384ed731-96a1-4c00-a830-345abfc827e2</item>
137   </items>
138 </view>
```

Next let's make some changes to the new article:

The screenshot shows a web browser window for Google Chrome with the URL <https://tm-tm4tm.azurewebsites.net/editor/384ed731-96a1-4c00-a830-345abfc827e2>. The page title is "TeamMentor". At the top right are three buttons: "Edit Source", "View Article", and "Save Changes". Below the title is a subtitle "a Security Innovation eKnowledge Project". A table has "Title" set to "Git Support for Libraries", "Technology" set to "Git", "Phase" to "Development", "Type" to "Version Control", and "Category" to "TM Architecture". Below the table is a rich text editor toolbar. The main content area contains the text: "TeamMentor supports the management of Libraries using Git. The best way to configure is to the Libraries_Git.Repositories field from the TMSecretData.config file".

After saving, the article is now available at

<https://tm-tm4tm.azurewebsites.net/article/384ed731-96a1-4c00-a830-345abfc827e2>²⁸ or
https://tm-tm4tm.azurewebsites.net/article/Git_Support_for_Libraries²⁹

²⁸<https://tm-tm4tm.azurewebsites.net/article/384ed731-96a1-4c00-a830-345abfc827e2>

²⁹https://tm-tm4tm.azurewebsites.net/article/Git_Support_for_Libraries

The screenshot shows a web browser window for 'Git Support for Libraries - TeamMentor 3.3 - Google Chrome'. The URL is <https://tm-tm4tm.azurewebsites.net/article/384ed731-96a1-4c00-a830-345abfc827e2>. The page features the TeamMentor logo and navigation buttons for 'Edit WYSIWYG' and 'Edit Source'. A table lists 'Technology' (Git), 'Category' (TM Architecture), 'Phase' (Development), and 'Type' (Version Control). Below the table, text states: 'TeamMentor supports the management of Libraries using Git. The best way to configure is to the the Libraries_Git.Repositories field from the TMSecretData.config file'. At the bottom, it says 'TeamMentor © 2007-2013 all rights reserved - a [Security Innovation](#) eKnowledge Product | [Contact Us](#)'.

As before, there is a new Commit at GitHub (https://github.com/TMContent/Lib_TM4TM/commits/³⁰):

The screenshot shows the GitHub commit history for the repository 'Lib_TM4TM'. It displays two commits made on March 29, 2013. The first commit, authored by an unknown user 4 minutes ago, changed the file 'Articles/384ed731-96a1-4c00-a830-345abfc827e2.xml'. The second commit, also by an unknown user 10 minutes ago, added the same file. Both commits have their commit hash (802065db6d and efdd3c9acf) and a 'Browse code' link.

Which contains the ‘metadata changes’ and the new article’s html content (created by the WYSIWYG TM online editor):

³⁰https://github.com/TMContent/Lib_TM4TM/commits/

The screenshot shows a GitHub commit interface for a file named 'Articles/384ed731-96a1-4c00-a830-345abfc827e2.xml'. The commit was authored by 'unknown-user' 4 minutes ago. It has 7 additions and 5 deletions across 12 lines. The diff shows changes in the XML structure, including the addition of a 'Title' element ('New Guidance Item') and the addition of a 'Content' section containing a CDATA block with the text 'TeamMentor supports the management of Libraries using Git.' followed by a descriptive paragraph.

```

changed: Articles/384ed731-96a1-4c00-a830-345abfc827e2.xml
unknown-user authored 4 minutes ago
1 parent efd3c9 commit 80f065db6d00997dd10c20ef537406baa73dd684
Showing 1 changed file with 7 additions and 5 deletions.
View file @ 80f065d
12  Articles/384ed731-96a1-4c00-a830-345abfc827e2.xml
...
3 ... @@ -3,15 +3,17 @@
4   <Metadata>
5     <Id>384ed731-96a1-4c00-a830-345abfc827e2</Id>
6     <Library_Id>13e0ebf0-2064-4031-9be7-3de4537ed3a9</Library_Id>
7     - <Title>New Guidance Item</Title>
8     - <Category />
9     - <Technology />
10    - <Type />
11    + <Title>Git Support for Libraries</Title>
12    + <Category>TM Architecture</Category>
13    + <Phase>Development</Phase>
14    + <Technology>Git</Technology>
15    + <Type>Version Control</Type>
16    + <DirectLink>Git Support for Libraries</DirectLink>
17    <Author />
18    <Priority />
19    <Status />
</Metadata>
<Content Sanitized="false" DataType="html">
- <Data><![CDATA[]]></Data>
+ <Data><![CDATA[TeamMentor supports the management of Libraries using Git.<div><br /></div>The best way to
</Content>
</TeamMentor_Article>

```

And since this is all Git based, many more complex and multi-user/hosting scenarios are easily supported (for example I can have a local copy of the TM4TM server/repo which I can edit offline and push to GitHub (directly or via Pull Requests)).

The git merge strategy is the same used by GitHub:

- If there are no conflicting changes, everything happens automatically or via GUIs pages
- If there are merge conflicts, the Git Bash and Windows Diff tools should be used to address them

6. April 2013

- Git pulling a TeamMentor Library and renaming it
- Creating QA versions of TeamMentor UserData repository, and using branches to show/test the multiple config options
- Changing a User's ExpiryDate from GitHub hosted file
- Linus gift to the world will be Git not Linux (and what about an OS built on top of an hash-driven file system?)
- What the move from HTML to WikiText looks like (in GitHub)
- Is Git a Single point of failure for TeamMentor?
- Setting up Ian's CI Development Environment (for TeamMentor)

Git pulling a TeamMentor Library and renaming it

Here is an example of how to use the new TM 3.3 capabilities to load libraries from GitHub and to rename them.

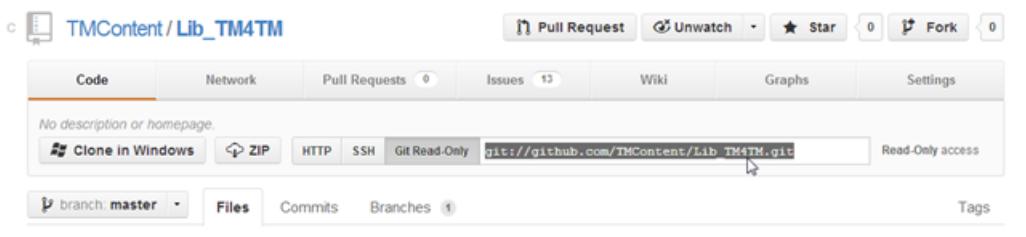
Let's start with a version of TM that looks like this:

The screenshot shows the TeamMentor application running on a local host at port 3187. The main interface includes a navigation bar with back, forward, and search icons, and a URL bar showing 'localhost:3187/teamMentor'. On the right, there are links for 'Login' and 'About Help'. The main content area has a search bar and two filter panels: 'Technology' (with 'Any' and 'C++' options) and 'Phase' (with 'Deployment', 'Design', 'Implementation', and 'Test' options). Below these filters, a message says 'Showing 489 items (out of 489)'. To the right, a 'Selected Article' panel displays a 'Login required' message with fields for 'Username' and 'Password'. A sidebar on the left lists 'Applied Filters' and 'Guidance Libraries' under the 'C++' category, which includes 'Fundamentals of Security', 'PCI DSS Code Review', 'PCI DSS Compliance', 'Security Engineering', and 'D+AAA'.

And let's say that we wanted to add the TM4TM Library to this server

The screenshot shows a GitHub repository page for 'TMContent / Lib_TM4TM'. The top navigation bar includes 'Pull Request', 'Unwatch', 'Star', 'Fork', and 'Settings'. Below the header, there are tabs for 'Code', 'Network', 'Pull Requests', 'Issues', 'Wiki', 'Graphs', and 'Settings'. The 'Code' tab is selected, showing a message 'No description or homepage.' and a clone button for 'Windows'. Below this, there are buttons for 'ZIP', 'HTTP', 'SSH', 'Git Read-Only', and a copy link 'git@github.com:TMContent/Lib_TM4TM.git'. A note indicates 'Read-Write access'. Under the code tab, there are dropdowns for 'branch: master' and 'Files', and tabs for 'Commits', 'Branches', and 'Tags'. The 'Commits' tab is selected, showing a list of 86 commits. The first commit by 'Roman87' is dated 16 hours ago and changes 'TM4TM.xml'. Subsequent commits include 'Articles', 'Content Writing Process', 'Development Process', 'Guidance Articles', 'QA', and 'Images' with various file additions and changes.

First thing to do is to copy the Git's *Read-Only Url*

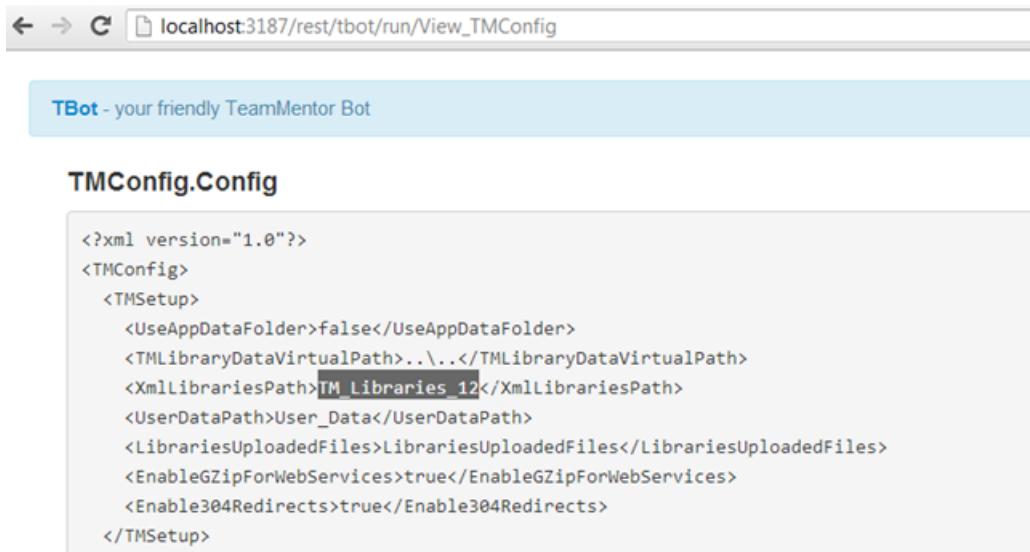


And add it to the TBot's Secret Data file:

```
{
  "Libraries_Git.Repositories": [
    {
      "url": "git://github.com/TMContent/Lib_TM4TM.git"
    }
  ],
  "SMTP_Password": null,
  "SMTP_Server": "smtp.sendgrid.net",
  "SMTP_UserName": "TeamMentor"
}
```

Before we reload the cache (which will do the git pull using NGit), lets see what the Library's folder looks like.

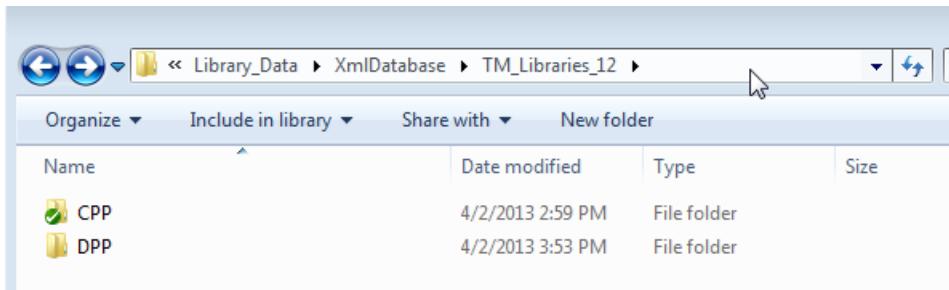
In this instance of TM, as we can see by the TMConfig.config file:



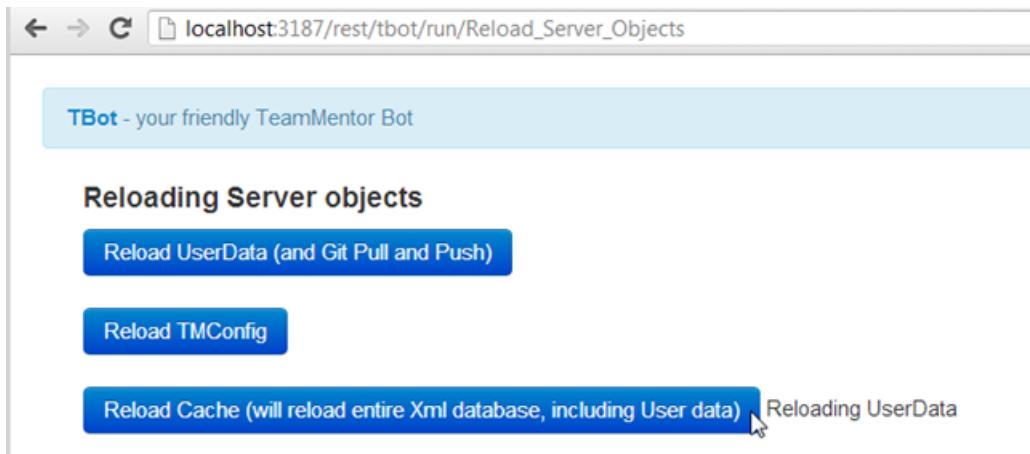
The screenshot shows a web browser window with the URL `localhost:3187/rest/tbot/run/View_TMConfig`. The page title is "TBot - your friendly TeamMentor Bot". Below it, the section title is "TMConfig.Config". The content is an XML configuration file:

```
<?xml version="1.0"?>
<TMConfig>
  <TMSetup>
    <UseAppDataFolder>false</UseAppDataFolder>
    <TMLibraryDataVirtualPath>...\\</TMLibraryDataVirtualPath>
    <XmlLibrariesPath>IM_Libraries_12</XmlLibrariesPath>
    <UserDataPath>User_Data</UserDataPath>
    <LibrariesUploadedFiles>LibrariesUploadedFiles</LibrariesUploadedFiles>
    <EnableGZipForWebServices>true</EnableGZipForWebServices>
    <Enable304Redirects>true</Enable304Redirects>
  </TMSetup>
```

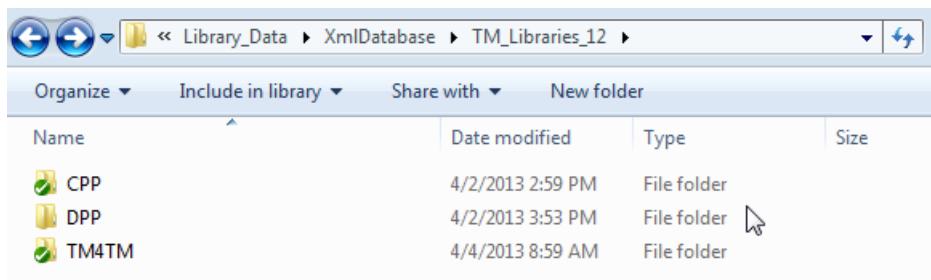
The Library files are located in the TM_Libraries_12 folder:



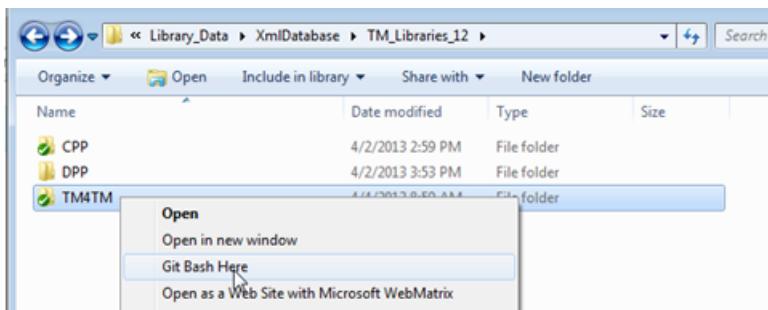
And if we now trigger the cache reload:



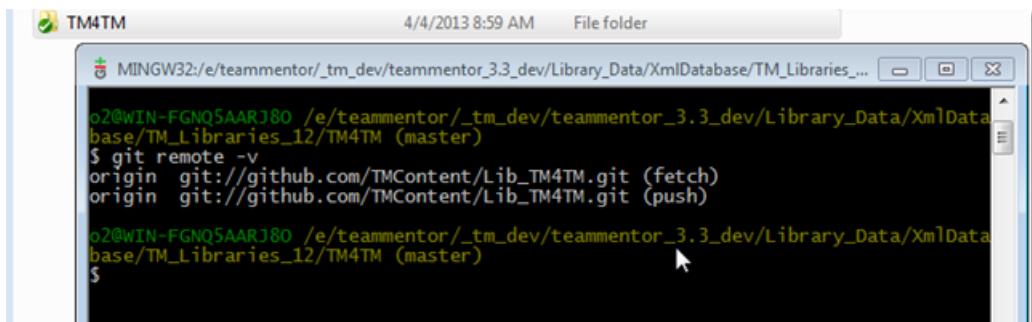
We will see that there is a new TM4TM folder:



which is a git repository



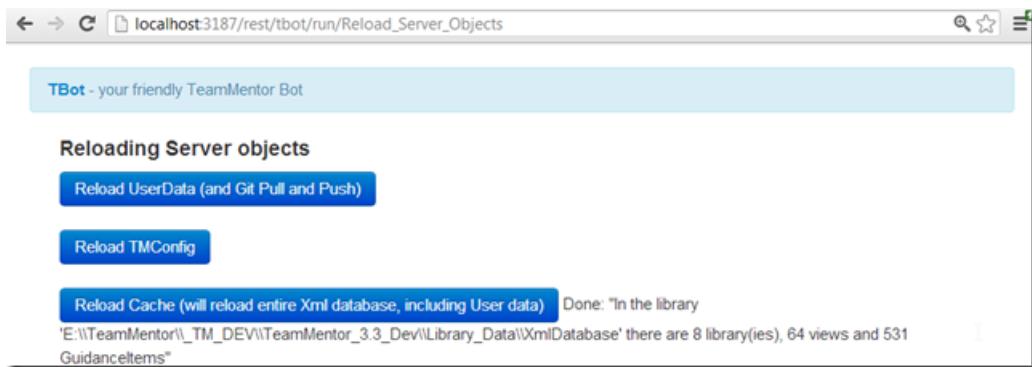
with its remote set to the Git's **Read-Only Url**



```
o2@WIN-FGNQ5AARJ80 /e/teammentor/_tm_dev/teammentor_3.3_dev/Library_Data/XmlDatabase/TM_Libraries_12/TM4TM (master)
$ git remote -v
origin  git://github.com/TMContent/Lib_TM4TM.git (fetch)
origin  git://github.com/TMContent/Lib_TM4TM.git (push)

o2@WIN-FGNQ5AARJ80 /e/teammentor/_tm_dev/teammentor_3.3_dev/Library_Data/XmlDatabase/TM_Libraries_12/TM4TM (master)
$
```

After the cache reloads:



TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

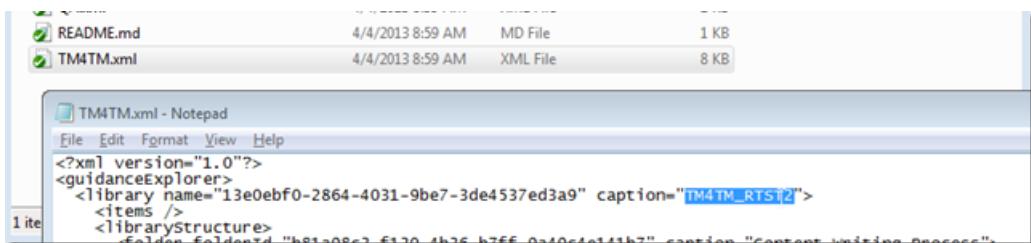
Reload Cache (will reload entire Xml database, including User data) Done: "In the library
'E:\TeamMentor__TM_DEV\TeamMentor_3.3_Dev\Library_Data\XmlDatabase' there are 8 library(ies), 64 views and 531 GuidanceItems"

There are now 8 Libraries loaded in TM:

The reason for the extra 6 libraries (when we only added one new repository) is that from TM 3.3, there can be more than one library file in library folder (note: the recommendation is have one library xml file per folder)

Name	Date modified	Type	Size
.git	4/4/2013 8:59 AM	File folder	
_Images	4/4/2013 8:59 AM	File folder	
Articles	4/4/2013 8:59 AM	File folder	
Content Writing Process	4/4/2013 8:59 AM	File folder	
Development Process	4/4/2013 8:59 AM	File folder	
Guidance Articles	4/4/2013 8:59 AM	File folder	
QA	4/4/2013 8:59 AM	File folder	
Content Writing Process.xml	4/4/2013 8:59 AM	XML File	3 KB
Development Process.xml	4/4/2013 8:59 AM	XML File	1 KB
Get Involved.xml	4/4/2013 8:59 AM	XML File	1 KB
Guidance Articles.xml	4/4/2013 8:59 AM	XML File	2 KB
QA.xml	4/4/2013 8:59 AM	XML File	1 KB
README.md	4/4/2013 8:59 AM	MD File	1 KB
TM4TM.xml	4/4/2013 8:59 AM	XML File	8 KB

Also note that the library name/caption is now independent from the xml file name:



Let's now open TM's Edit mode

Title	Technology	Phase
Additional Resources	TeamMentor	Article Development
Article Pair Example	TeamMentor	Article Development
Attack	TeamMentor	Article

and use it to rename the *TM4TM_RTST2* Library:

Technology	Phase
.Net	Article Creation
Browser	Article Development
Git	Contact
SSL	Development
TeamMentor	Implementation
	QA

from:

to:

The screenshot shows the TeamMentor application running in a web browser at localhost:3187/teamMentor. The main interface includes a navigation bar with links like 'Exit', 'Edit Mode', 'Change Password', 'Control Panel', and 'Logout'. A central search bar is present. On the left, there's a sidebar titled 'Applied Filters' and 'Guidance Libraries' containing a tree view of categories like C++, Fundamentals of Security, PCI DSS Code Review, etc. The main content area has two filter panels: 'Technology' (listing .Net, Browser, Git, SSL, TeamMentor) and 'Phase' (listing Article Creation, Article Development, Contact, Development, Implementation, QA). Below these filters, a message says 'Showing 42 items (out of 42)'. At the bottom of the main content area, there are buttons for 'Select All', 'Deselect All', 'Delete Guidance Items from Library', and 'Drag item t'. A large black rectangular box covers the top right corner of the main content area, containing the text 'Library Renamed'.

After the rename, a number of thinks happened.

1) The TM4TM.xml library file contents changed:

The screenshot shows a Windows File Explorer window. It lists two files: 'README.md' (MD File, 1 KB) and 'TM4TM.xml' (XML File, 8 KB). Below the file list is a Notepad window titled 'TM4TM.xml - Notepad'. The Notepad content displays the XML structure of the TM4TM library:

```

<?xml version="1.0"?>
<guidanceExplorer>
  <library name="13e0ebf0-2864-4031-9be7-3de4537ed3a9" caption="TM4TM">
    <items />
    <librarystructure>
      <folder folderId="b81a98c3-f130-4b36-b7ff-9a49c4e141b7" caption="Content Writing Process">
        <item itemid="13e0ebf0-2864-4031-9be7-3de4537ed3a9" caption="Content Writing Process" />
      </folder>
    </librarystructure>
  </library>
</guidanceExplorer>

```

2) There was a local commit with the change:

The screenshot shows the gitk interface with the following details:

- Local uncommitted changes, not checked in to index:**
 - master changed: TM4TM.xml
 - remotes/origin/master changed: TM4TM.xml
 - changed: Content Writing Process.xml , Development Process.xml , Get
 - Added: TM4TM.xml
 - removed: TM4TM.xml
 - Added: TM4TM.xml
 - removed: TM4TM.xml
 - changed: Articles/384ed731-96a1-4c00-a830-345abfc827e2.xml
- Commit Log (right side):**

Author	Date
DinisCruz <dinis.cruz@...>	2013-04-04 09:23:44
Roman87 <romich@...>	2013-04-03 17:09:35
Roman87 <romich@...>	2013-04-03 17:08:04
Roman87 <romich@...>	2013-04-02 12:13:08
Roman87 <romich@...>	2013-03-31 09:14:20
Roman87 <romich@...>	2013-03-30 15:37:09
Roman87 <romich@...>	2013-03-30 15:35:24
unknown-user <unkn...>	2013-03-30 02:01:31
- SHA1 ID:** f067925712d6f2144c669e82c3038eca0e5a3382
- Search:** Find next prev commit containing: Search
- Diff Options:** Diff Old version New version Lines of context: 3 Ignore space change Line diff
- Diff View:** TM4TM.xml


```
index b99a783..265dd81 100644
@@ -1,6 +1,6 @@
<?xml version="1.0"?>
<guidanceExplore>
- <library name="13e0ebf0-2864-4031-9be7-3de4537ed3a9" caption="TM4TM_RTST2">
+ <library name="13e0ebf0-2864-4031-9be7-3de4537ed3a9" caption="TM4TM">
  <name>
```
- Patch/Trees:** Patch Tree
- Comments:** TM4TM.xml

3) the auto pull to GitHub failed

This is confirmed by the commit list at GitHub:

The GitHub commit history shows two recent commits:

- Apr 03, 2013: Roman87 authored 16 hours ago. Commit message: changed: TM4TM.xml. SHA: 79ab28533f. Action: Browse code.
- Apr 03, 2013: Roman87 authored 16 hours ago. Commit message: changed: Content Writing Process.xml , Development Process.xml , Get SHA: b10caccd4. Action: Browse code.

and the ‘push error’ we got on the TBot’s *DebugInfo* page

The TBot’s DebugInfo page displays the following log output:

```
Logs (latest first)
Exception: Exception caught during execution of push command
INFO: [GitProgress] BeginTask : Opening connection : 0
DEBUG: [API_NGIt] push: origin
DEBUG: [API_NGIt] commit: changed: TM4TM.xml
```

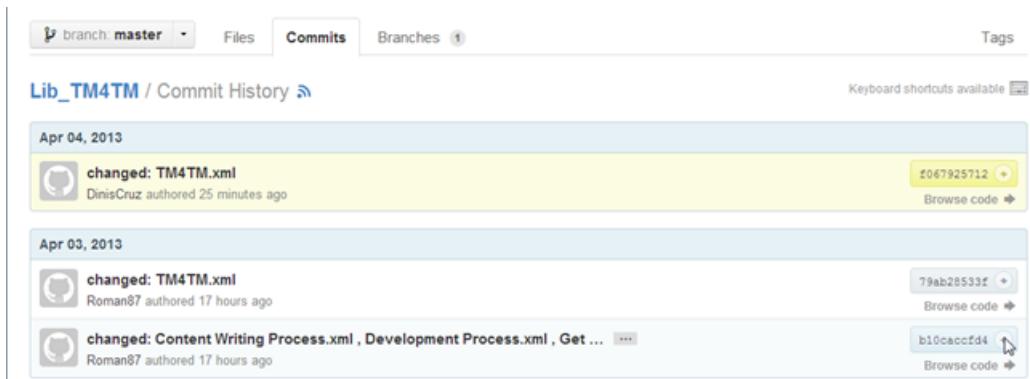
In this case I do want to push the changes, so back in GitHub I copied the **SSH** git url



And use it directly on a git push (I could also had done this by setting up a new remote)

```
o2@WIN-FGNQ5AARJ8O /e/teammentor/_tm_dev/teammentor_3.3_dev/Library_Data/XmlData
base/TM_Libraries_12(TM4TM (master)
$ git push git@github.com:TMContent/Lib_TM4TM.git
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 443 bytes, done.
Total 3 (delta 2), reused 0 (delta 0)
To git@github.com:TMContent/Lib_TM4TM.git
  79ab285..f067925  master -> master
```

Now the commit created by TM (on library rename) exists in GitHub:



Removing the extra Library files:

Since we don't need the extra libraries xml files, I just removed them (and committed the changes)

```

MINGW32:/e/teammentor/_tm_dev/teammentor_3.3_dev/Library_Data/XmlDatabase/TM_Libraries...
$ git add -A
o2@WIN-FGNQ5AARJ80 /e/teammentor/_tm_dev/teammentor_3.3_dev/Library_Data/XmlDatabase/TM_Libraries...
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       deleted:   Content Writing Process.xml
#       deleted:   Development Process.xml
#       deleted:   Get Involved.xml
#       deleted:   Guidance Articles.xml
#       deleted:   QA.xml
#
o2@WIN-FGNQ5AARJ80 /e/teammentor/_tm_dev/teammentor_3.3_dev/Library_Data/XmlDatabase/TM_Libraries...
$ git commit -m 'removing extra libraries'
[master 29204f2] removing extra libraries
 5 files changed, 121 deletions(-)
 delete mode 100644 Content Writing Process.xml
 delete mode 100644 Development Process.xml
 delete mode 100755 Get Involved.xml
 delete mode 100644 Guidance Articles.xml
 delete mode 100644 QA.xml
o2@WIN-FGNQ5AARJ80 /e/teammentor/_tm_dev/teammentor_3.3_dev/Library_Data/XmlDatabase/TM_Libraries...
$ git push git@github.com:TMContent/Lib_TM4TM.git
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 248 bytes, done.
Total 2 (delta 1), reused 0 (delta 0)
To git@github.com:TMContent/Lib_TM4TM.git
  f067925..29204f2 master -> master

```

9 items

Which means that after cache reload,

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, including User data) Done: "In the library
 'E:\TeamMentor_TM_DEV\TeamMentor_3.3_Dev\Library_Data\XmlDatabase' there are 3 library(ies), 51 views and 531 GuidanceItems"

there are now 3 libraries in my local TM instance:

The screenshot shows the TeamMentor web application running at localhost:3187/teamMentor. The interface includes a header with the TeamMentor logo and navigation links for Edit Mode, Change Password, Control Panel, Logout, TM 3.3 RC0 (3/April/2013), and a user log-in message. On the left, there's an 'Applied Filters' section and a 'Guidance Libraries' tree view with categories like C++, Fundamentals of Security, PCI DSS Code Review, PCI DSS Compliance, Security Engineering, D+AAAA, and TM4TM. The main content area features two filter panels: 'Technology' (listing .Net, Browser, Git, SSL, and TeamMentor) and 'Phase' (listing Article Creation, Article Develop..., Contact, Development, Implementation, and QA). Below these filters is a table titled 'Showing 42 items (out of 42)' with columns for Title, Technology, Phase, and Content. The first row in the table is highlighted in blue. The right side of the screen has a 'Selected Article' panel and an 'Additional Resources' panel containing links to various TeamMentor resources and documentation.

localhost:3187/teamMentor

TeamMentor
a Security Innovation eKnowledge Product

Applied Filters

Guidance Libraries

- C++
 - Fundamentals of Security
 - PCI DSS Code Review
 - PCI DSS Compliance
 - Security Engineering
- D+AAAA
- TM4TM

Technology

- .Net
- Browser
- Git
- SSL
- TeamMentor

Phase

- Article Creation
- Article Develop...
- Contact
- Development
- Implementation
- QA

Showing 42 items (out of 42)

Title	Technology	Phase	Content
Additional Resources	TeamMentor	Article Development	Content
Article Pair Example	TeamMentor	Article Development	Content
		Article	Content

Selected Article

Additional Resources

For the current version of TeamMentor content, see <http://teammentor.net>
For a public OWASP TeamMentor library, see <http://owasp.teammentor.net>
For TeamMentor UI documentation, see <http://docs.teammentor.net>
For TeamMentor schemas and other information about writing the articles, see <http://tm4tm.teammentor.net/teamMentor>

TeamMentor © 2007-2013 all rights reserved | a [Security Innovation](#) eKnowledge Product | [Contact Us](#)

Creating QA versions of TeamMentor UserData repository, and using branches to show/test the multiple config options

Now that a number of TeamMentor settings can be configured from the UserData repositories, we need a way to test and document what can be done.

Let's start by creating a public GitHub repository (https://github.com/TeamMentor/UserData_Customizations¹) to hold the multiple examples/tests:

The screenshot shows a GitHub repository page for 'TeamMentor / UserData_Customizations'. The repository is public. At the top, there are buttons for 'Pull Request', 'Unwatch', 'Star', 'Fork', and 'Settings'. Below the header, there are links for 'Code', 'Network', 'Pull Requests (0)', 'Issues (0)', 'Wiki', 'Graphs', and 'Settings'. A note says 'QA examples of the customizations that can be done using the UserData repository — [Read more](#)'. Below this, there are buttons for 'Clone in Windows', 'ZIP', 'HTTP', 'SSH', 'Git Read-Only', and 'git@github.com:TeamMentor/UserData_Customizations'. A 'Read+Write access' button is also present. The main content area shows a commit history. One commit is shown, made by 'DinisCruz' a day ago, updating 'README.md'. The commit hash is '0d9bfee20f'. Below the commit, there is a section titled 'README.md' with the heading 'Lib_QA_UserData'. It contains the text: 'This repository has multiple QA examples of the customizations that can be done using the UserData repository.' and 'Checkout the multiple branches for examples.'

Set the GitUser location to it:

¹https://github.com/TeamMentor/UserData_Customizations

TBot - your friendly TeamMentor Bot

Editing an Secret Data

Git User Location:

Save data saved: true

And reload the user data:

localhost:3187/rest/tbot/run/Reload_Server_Objects

TBot - your friendly TeamMentor Bot

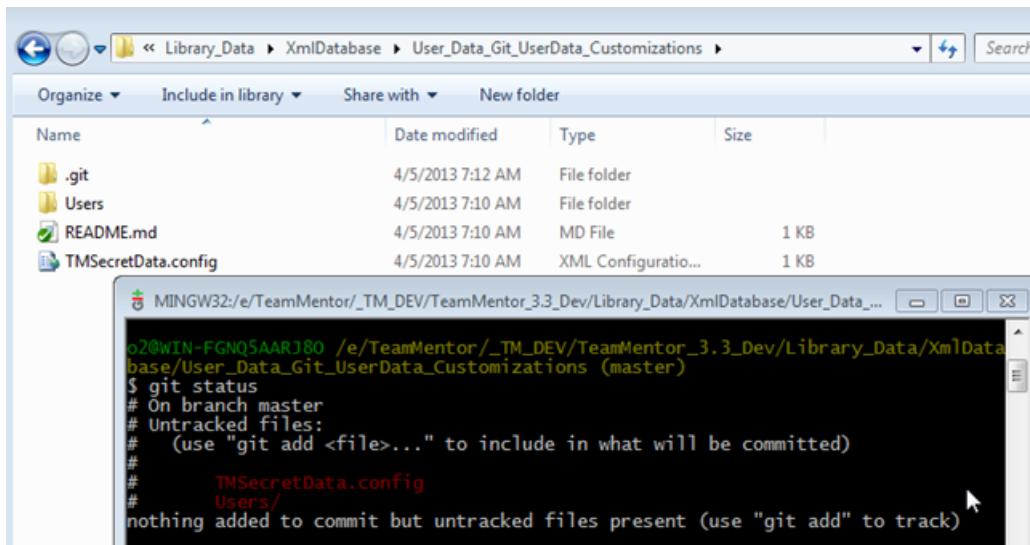
Reloading Server objects

Reload UserData (and Git Pull and Push) Done: true

Once the reload is complete, there will be a new folder called **User_Data_Git_UserData-Customizations** in the local XmlDatabase folder:

Name	Date modified	Type	Size
.git	4/5/2013 7:10 AM	File folder	
Users	4/5/2013 7:10 AM	File folder	
README.md	4/5/2013 7:10 AM	MD File	1 KB
TMSecretData.config	4/5/2013 7:10 AM	XML Configuration...	1 KB

Note that if you are running TM from localhost (as in the current example) then the user data will not be auto committed (due to the dynamic nature of UserData, if GitAutoCommit was enabled it would not be possible to load userdata repositories used on live TM sites (like the multiple *Site_nnn.git* ones) without creating commit conflicts):



Before we move to the branches let's commit the current TMSecretData.config and admin files:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_Data_Git_UserData_Customizations (master)
$ git add .

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_Data_Git_UserData_Customizations (master)
$ git commit -m 'first commit of TMSecretData and admin user'
[master a646987] first commit of TMSecretData and admin user
 2 files changed, 14 insertions(+)
  create mode 100644 TMSecretData.config
  create mode 100644 Users/admin_42e421a3-72fd-4e38-bbdd-ada7e2edd8fe.userData.xml
```

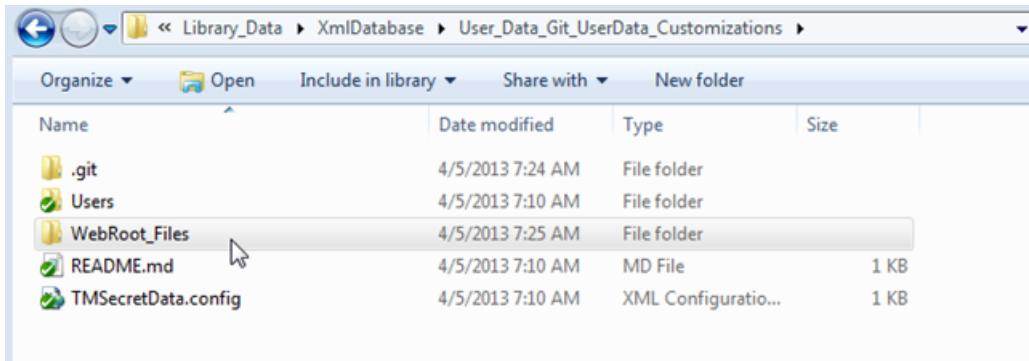
Use Case #1: Changing version by running Customized Javascript code

The first example is going to show how to execute some Javascript in the main TM Gui from a file provided in the user data folder.

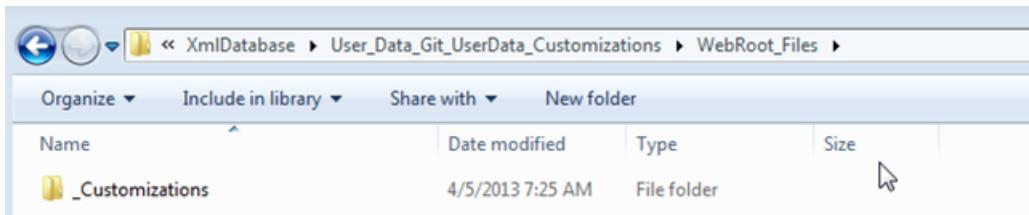
Let's create a branch to hold the changes:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_Data_Git_UserData_Customizations (master)
$ git checkout -b UseCase_Javascript_Customization
Switched to a new branch 'UseCase_Javascript_Customization'
```

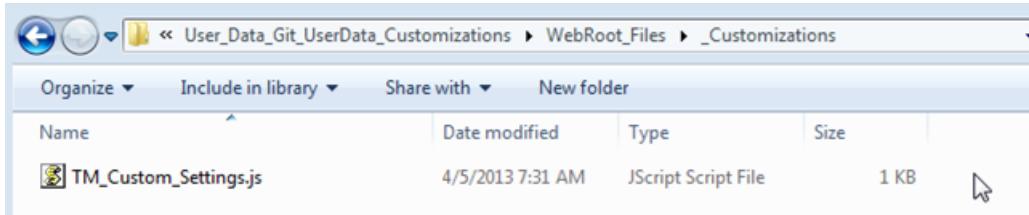
Add a folder called ***WebRoot_Files***:



Add a folder called **_Customizations** (inside the **WebRoot_Files**)



Add a JavaScript file called **TM_Custom_Settings.js** inside the **_Customizations** folder:



Note: the reason for this file, is that it is automatically included (if it exists) in the Javascript consolidated download that is done on the main TM GUI. Here is the mapping file that also shows the execution order of this script:

```

HomePage_JS_TM.txt
-----
/javascript/TM/GlobalVariables.js
/javascript/TM/GuiObjects.js
/javascript/TM/Settings.js
/javascript/TM/Events.js
/javascript/TM/Debug.js

/_Customizations/TM_Custom_Settings.js

/javascript/IE_Fixes.js

/javascript/j02/j02_String_ExtensionMethods.js
/javascript/j02/j02_jQuery_ExtensionMethods.js

/javascript/jQueryUI/TM.jQueryUI.ExtraMethods.js
/javascript/jQuery.DataTable/DataTable.ExtraMethods.js

/javascript/TM.Gui/TM.GUI.ShowProgressBar.js

/javascript/TM_HelperMethods.js

/javascript/TM/WebServices.js
/javascript/TM_WS_Methods.js

/javascript/TM.Gui/TM.GUI.LibraryTree.js
/javascript/TM.Gui/TM.GUI.Dialog.js
/javascript/TM.Gui/TM.GUI.ShowProgressBar.js
/javascript/TM.Gui/TM.GUI.Main.js
/javascript/TM.Gui/TM.Gui.Main.Panels.js
/javascript/TM.Gui/TM.Gui.CurrentUser.js
/javascript/TM.Gui/TM.Gui.DataTable.js
/javascript/TM.Gui/TM.Gui.DataTableViewer.js
/javascript/TM.Gui/TM.Gui.AppliedFiltersList.js

/javascript/TM.Gui/AppliedFilters.js
/javascript/TM.Gui/AppliedFilters.DataManipulation.js

/gAnalytics/TrackingCode.js
/gAnalytics/ga.js

```

Next edit the *TM_Custom_Settings.js* file and use it to (for example) change the *TM.tmVersion* value.

```

TM_Custom_Settings.js
-----
TM.tmVersion = TM.tmVersion + " (UseCase_Javascript_Customization)";

```

And in TBot , trigger a *Cache Reload*

TBot - your friendly TeamMentor Bot

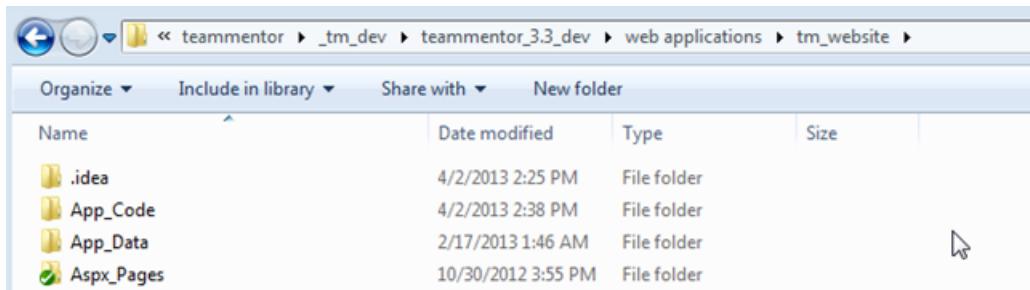
Reloading Server objects

Reload UserData (and Git Pull and Push) Done: true

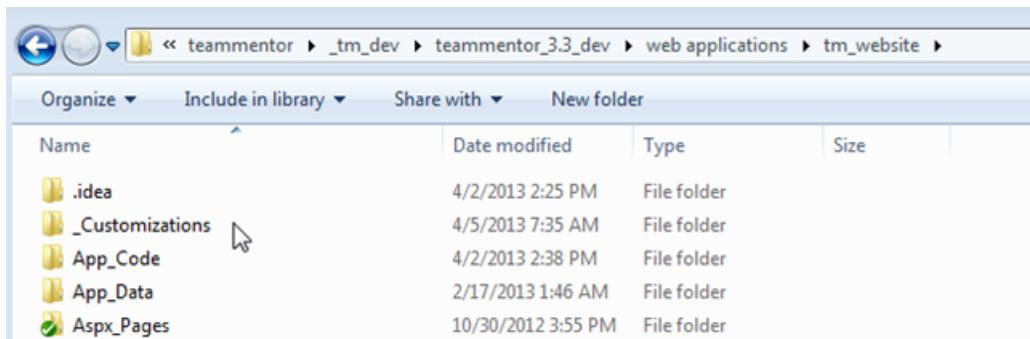
Reload TMConfig Done: true

Reload Cache (will reload entire Xml database, including User data) Done: "In the library"

If you keep an eye in the TM_WebSite folder (the root of the TM website), you will see that it looks like this before the the **Cache Reload**

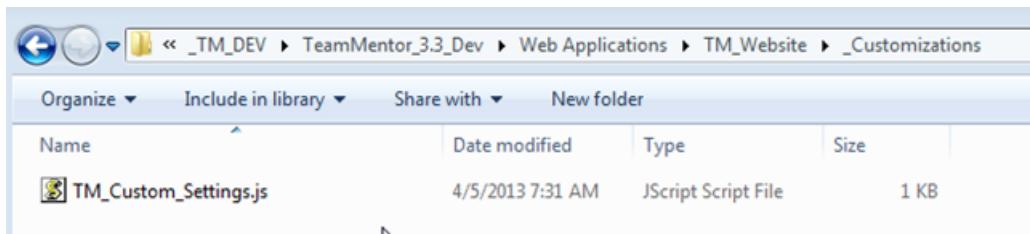


and like this after the **Cache Reload**:

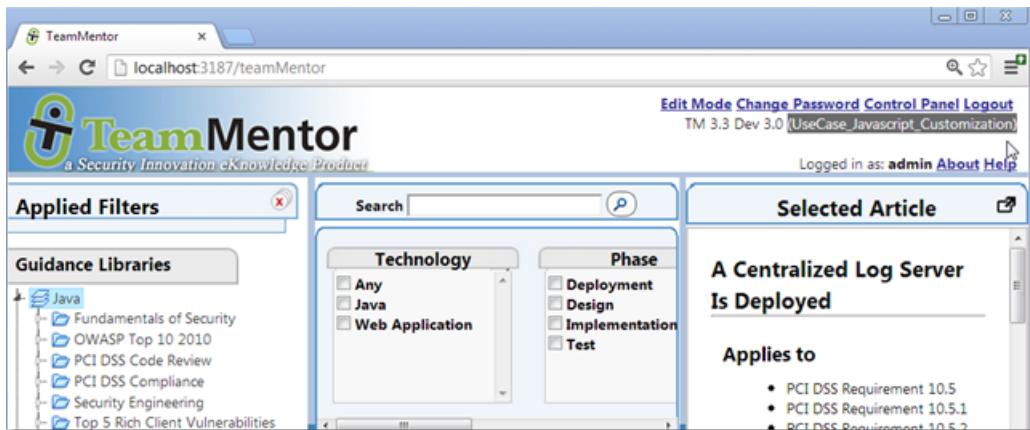


What happened is that the contents of the UserData's WebRoot_Files was copied into TM's web root.

Which means that the **TM_Custom_Settings.js** created above is now part of TM:



A hard refresh of the browser, will now show the customized ***TM.tmVersion*** value:



To wrap up this branch, let's modify the README.md file:

```
README.md - Notepad
File Edit Format View Help
User_Data_Git_UserData_Customizations
=====
This repository has multiple QA examples of the customizations that can be done using the userdata repository.
Checkout the multiple branches for examples
Current Branch: UseCase_Javascript_Customization
=====
Simple example of how to add some Javascript code to TeamMentor's main gui.
In this case, adding the value (useCase_Javascript_Customization) to the TM.tmversion javascript variable (show
```

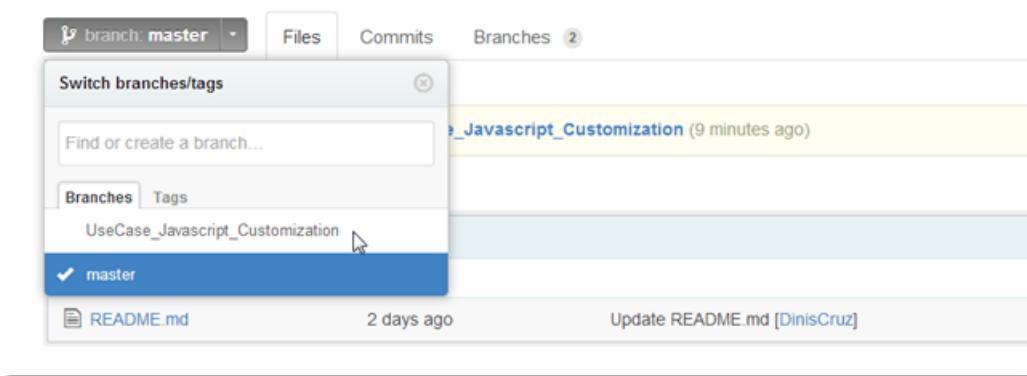
Commit the changes:

```
$ git commit -m 'adding example for UseCase_Javascript_Customization'
[UseCase_Javascript_Customization 34162ab] adding example for UseCase_Javascript_Customization
2 files changed, 8 insertions(+), 2 deletions(-)
create mode 100644 WebRoot_Files/_Customizations/TM_Custom_Settings.js
```

And push the branch to GitHub (note the explicit branch mapping on the git push command):

```
02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_Data_Git  
ns (UseCase_Javascript_Customization)  
$ git push origin UseCase_Javascript_Customization:UseCase_Javascript_Customization  
Counting objects: 13, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (10/10), done.  
Writing objects: 100% (12/12), 1.91 KiB, done.  
Total 12 (delta 2), reused 0 (delta 0)  
To git@github.com:TeamMentor/UserData_Customizations.git  
 * [new branch]      UseCase_Javascript_Customization -> UseCase_Javascript_Customization
```

A quick look at GitHub's repo:



will show our branch and modified files:

The screenshot shows a GitHub repository named 'UserData_Customizations'. At the top, there are buttons for 'Pull Request' and 'Compare', and the text 'UseCase_Javascript_Customization (10 minutes ago)'. Below this, the repository name is followed by a link and a '4 commits' badge. A table lists four commits made by 'DinisCruz' in the last 35 minutes, with the latest commit being '34162abec2'. The commits are: 'adding example for UseCase_Javascript_Customization' (Users), 'adding example for UseCase_Javascript_Customization' (WebRoot_Files), 'Update README.md' (README.md), and 'first commit of TMSecretData and admin user' (TMSecretData.config). Below the commits is a 'README.md' file containing the following content:

```
Lib_QA_UserData

This repository has multiple QA examples of the customizations that can be done using the UserData repository.

Checkout the multiple branches for examples
```

Use Case #2: Setting Google Analytics (server-side) value by running Customized C# code

In this example we will set enabled and configure the Server side google analytics settings (used to provide metrics on TM usage)

Note: see the [Running Customized C# code loaded from TeamMentor's UserData repository²](#) post to understand the role of the *FirstScriptToInvoke.h2* script

Let's create a new branch called **UseCase_CSharp_Customization** using as a starting point the existing **UseCase_Javascript_Customization**

```
o7@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XMLDatabase>
ns (UseCase_Javascript_Customization)
$ git checkout -b UseCase_CSharp_Customization
Switched to a new branch 'UseCase_CSharp_Customization'

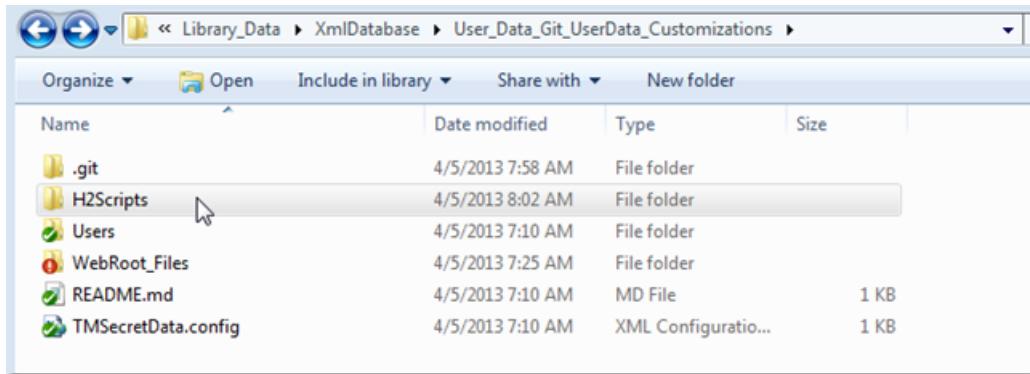
o7@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XMLDatabase>
ns (UseCase_CSharp_Customization)
$
```

Modify the **WebRoot_Files_CustomizationsTM_Custom_Settings.js** file:

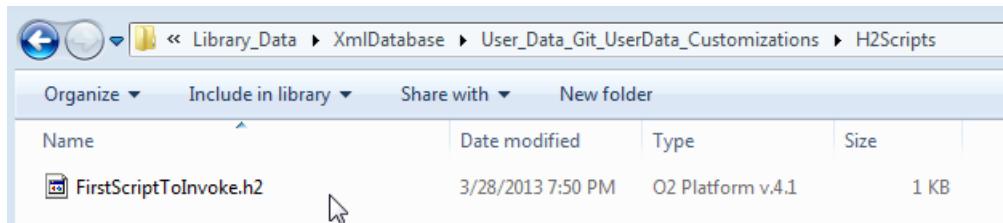
²<http://blog.diniscruz.com/2013/04/running-customized-c-code-loaded-from.html>

```
TM_Custom_Settings.js
TM.tmVersion = TM.tmVersion + " (UseCase_CSharp_Customization);
```

In the User data repository (which is checkout into the *UseCase_CSharp_Customization* branch), add the *H2Scripts* folder



Inside that folder add the *FirstScriptToInvoke.h2* file



Which (as explained in this post³) will be executed on startup.

Before adding code to the *FirstScriptToInvoke.h2* file, its good to test/debug that code using the TM's CSharp REPL:

³<http://blog.diniscruz.com/2013/04/running-customized-c-code-loaded-from.html>

Web C# REPL is a simple wrapper on O2 Platform's FluentSharp.CoreLib API
Note that all your script executions are logged (see link on top) so don't put any sensitive data here ;)

```

1 var googleAnalytics          = GoogleAnalytics.Current;
2 googleAnalytics.AccountID    = "UA-37594728-3";
3 googleAnalytics.Enabled      = true;
4 googleAnalytics.LogWebServicesCalls = true;
5
6 return googleAnalytics;
7
8 //using TeamMentor.CoreLib;
9 //O2Ref:TeamMentor.CoreLib.dll

```

and confirm on the log viewer that Google Analytics (GA) calls are now being logged:

Log Viewer

[9:01:32 AM] INFO: [GA] [TM_Log] : WebService -> http://localhost:3187/aspx_pages/TM_WebServices.asmx/REPL_ExecuteScript
[9:00:57 AM] INFO: [GA] [TM_Log] : REST -> http://localhost:3187/rest/tbot/run/DebugInfo

Once we're happy with the C# snippet to execute, we can add it to the *FirstScriptToInvoke.h2* file:

HandleUrlRequest.cs FirstScriptToInvoke.h2

```

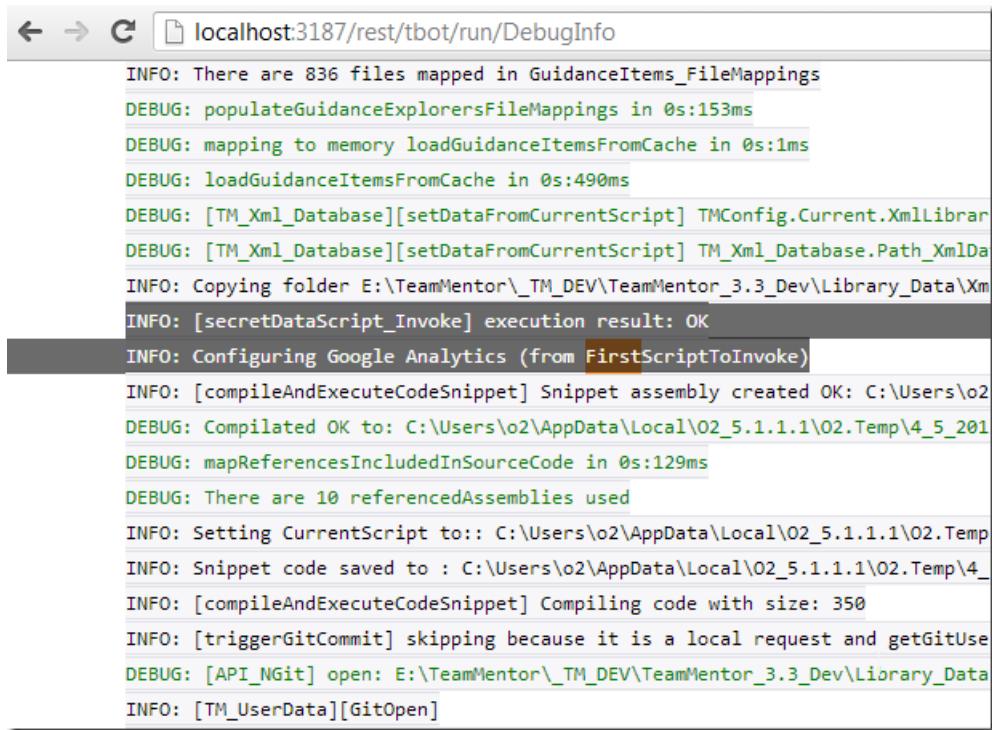
"Configuring Google Analytics (from FirstScriptToInvoke)".info();
var googleAnalytics          = GoogleAnalytics.Current;
googleAnalytics.AccountID    = "UA-37594728-3";
googleAnalytics.Enabled      = true;
googleAnalytics.LogWebServicesCalls = true;

return "OK";

//using TeamMentor.CoreLib;
//O2Ref:TeamMentor.CoreLib.dll

```

To double check that the Google Analytics (GA) settings are being configured on TM setup, I restarted Cassini, and confirmed that the log viewer shows the *FirstScriptToInvoke.h2* file execution:



The screenshot shows a browser window with the URL `localhost:3187/rest/tbot/run/DebugInfo`. The page content displays a log of system messages. A specific message, "INFO: Configuring Google Analytics (from FirstScriptToInvoke)", is highlighted with a dark gray background and white text.

```
INFO: There are 836 files mapped in GuidanceItems_FileMappings
DEBUG: populateGuidanceExplorersFileMappings in 0s:153ms
DEBUG: mapping to memory loadGuidanceItemsFromCache in 0s:1ms
DEBUG: loadGuidanceItemsFromCache in 0s:490ms
DEBUG: [TM_Xml_Database][setDataFromCurrentScript] TMConfig.Current.XmlLibrary
DEBUG: [TM_Xml_Database][setDataFromCurrentScript] TM_Xml_Database.Path_XmlDa
INFO: Copying folder E:\TeamMentor\_TM_DEV\TeamMentor_3.3_Dev\Library_Data\Xm
INFO: [secretDataScript_Invoke] execution result: OK
INFO: Configuring Google Analytics (from FirstScriptToInvoke)
INFO: [compileAndExecuteCodeSnippet] Snippet assembly created OK: C:\Users\o2
DEBUG: Compilated OK to: C:\Users\o2\AppData\Local\O2_5.1.1.1\O2.Temp\4_5_201
DEBUG: mapReferencesIncludedInSourceCode in 0s:129ms
DEBUG: There are 10 referencedAssemblies used
INFO: Setting CurrentScript to:: C:\Users\o2\AppData\Local\O2_5.1.1.1\O2.Temp
INFO: Snippet code saved to : C:\Users\o2\AppData\Local\O2_5.1.1.1\O2.Temp\4_
INFO: [compileAndExecuteCodeSnippet] Compiling code with size: 350
INFO: [triggerGitCommit] skipping because it is a local request and getGitUse
DEBUG: [API_NGIt] open: E:\TeamMentor\_TM_DEV\TeamMentor_3.3_Dev\Library_Data
INFO: [TM_UserData][GitOpen]
```

and the successful configuration/use of Google Analytics:



The screenshot shows a browser window with the URL `localhost:3187/rest/tbot/run/DebugInfo`. The page content displays a log of system messages. A section titled "Logs (latest first)" is shown, containing two log entries:

INFO: [GA] [TM_Log] : REST -> http://localhost:3187/rest/tbot/run/DebugInfo
INFO: [GA] [TM_Log] : REST -> http://localhost:3187/rest/tbot/run/Commands

Final step is to update the README.md file

README.md - Notepad

File Edit Format View Help

User_Data_Git_UserData_Customizations

=====

This repository has multiple QA examples of the customizations that can be done using the UserData repository.

Checkout the multiple branches for examples

Current Branch: UseCase_Csharp_Customization

=====

Simple example of how to add some C# code that will be executed when TeamMentor starts

In this case configuring the Google Analytics server side data

Add the files to git

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_Data_Git
ns (UseCase_CSharp_Customization)
$ git add .

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_Data_Git
ns (UseCase_CSharp_Customization)
$ git status
# On branch UseCase_CSharp_Customization
# Changes to be committed:
#   (use "git reset HEAD <file>" to unstage)
#
#       new file:   H2Scripts/FirstScriptToInvoke.h2
#       modified:  README.md
#       modified:  Users/admin_42e421a3-72fd-4e38-bbdd-ada7e2edd8fe.userData.xml
#       modified:  WebRoot_Files/_Customizations/TM_Custom_Settings.js
```

Commit the changes:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_
ns (UseCase_CSharp_Customization)
$ git commit -m 'adding example for UseCase_CSharp_Customization'
[UseCase_CSharp_Customization d5f4ae1] adding example for UseCase_CSharp_Customization
 4 files changed, 47 insertions(+), 2 deletions(-)
  create mode 100644 H2Scripts/FirstScriptToInvoke.h2
```

And push Commit into a new branch at GitHub:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev/Library_Data/XmlDatabase/User_
ns (UseCase_CSharp_Customization)
$ git push origin UseCase_CSharp_Customization:UseCase_CSharp_Customization
Counting objects: 17, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 1.28 KiB, done.
Total 10 (delta 1), reused 0 (delta 0)
To git@github.com:TeamMentor/UserData_Customizations.git
 * [new branch]      UseCase_CSharp_Customization -> UseCase_CSharp_Customization
```

Changing a User's ExpiryDate from GitHub hosted file

For the cases where TeamMentor UserData is loaded from a GitHub repository, it is possible to change/manage user data directly from GitHub's web GUI (or from a local clone of that repository).

Lets take for example Danny's account, which is expired at the moment (today is 4/10/2013):

TBot - your friendly TeamMentor Bot

[View User](#) [Edit User](#) [View Activity/Logs](#) [Raw/Xml Data](#)

User details

User Name: danny
Email: dharris@securityinnovation.com
First Name: Danny
Last Name: Harris
Company: SI
Job/Title: No idea
County: US
State: One of them
GroupID: 3
Account Expires: 4/5/2013 3:59:32 AM 
Account Enabled: True
PasswordExpired: False

Stats

Creation Date: 3/21/2013 3:59:32 AM
Last Login: 4/4/2013 2:39:20 PM
Login Ok: 3
Login Fail: 0

In GitHub, this is the file that contains Danny's user data:

Site_tm4tm.teammentor.net / Users / [+](#)

removed: Users/dinis_c3354a39-9dca-475c-9ca2-3b536cba4b04.userData.xml

unknown-user authored 3 minutes ago

..

DinisCruz_4262d4f7-0102-4bd1-ae03-31c1...	4 minutes ago	changed: Users/DinisCruz_4262d4f7-0102-4bd1
admin_e391b43c-7c69-4de5-ba3a-10ea06...	4 minutes ago	changed: Users/admin_e391b43c-7c69-4de5-ba
danny.harr_50feb19c-7059-430c-95fe-103...	20 days ago	changed: Users/danny.harr_50feb19c-7059-430
danny_17138152-f11e-4573-aeb8-227db7...	5 days ago	changed: Users/danny_17138152-f11e-4573-ae

So we open and edit that file:

Site_tm4tm.teammentor.net / Users / danny_17138152-f11e-4573-aeb8-227db7e2735b.userData.xml

unknown-user 5 days ago changed: Users/danny_17138152-f11e-4573-aeb8-227db7e2735b userData.xml

0 contributors

File | 69 lines (69 sloc) | 10.15 kb Edit Raw Blame History

```

1 <?xml version="1.0"?>
2 <TMUser ID="17138152-f11e-4573-aeb8-227db7e2735b" UserID="1816626860"
  ...
  
```

Change the *ExpirationDate* to a value in 2014

Site_tm4tm.teammentor.net / Users / [danny_17138152-f11e-4!](#) or car

Code Preview

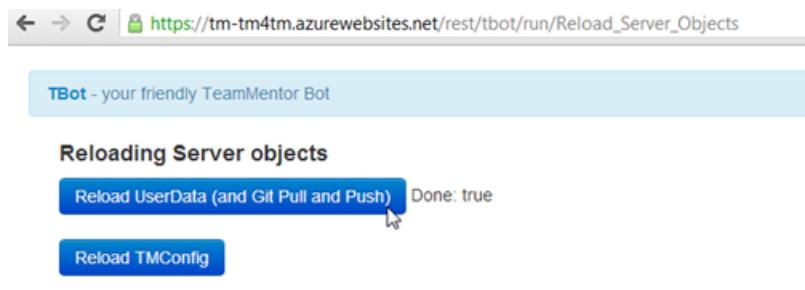
```

1 <?xml version="1.0"?>
2 <TMUser ID="17138152-f11e-4573-aeb8-227db7e2735b" UserID="1816626860"
  ...
  
```

Commit the changes:



Reload the UserData:



And the Danny account details at the server is now set to the new date:

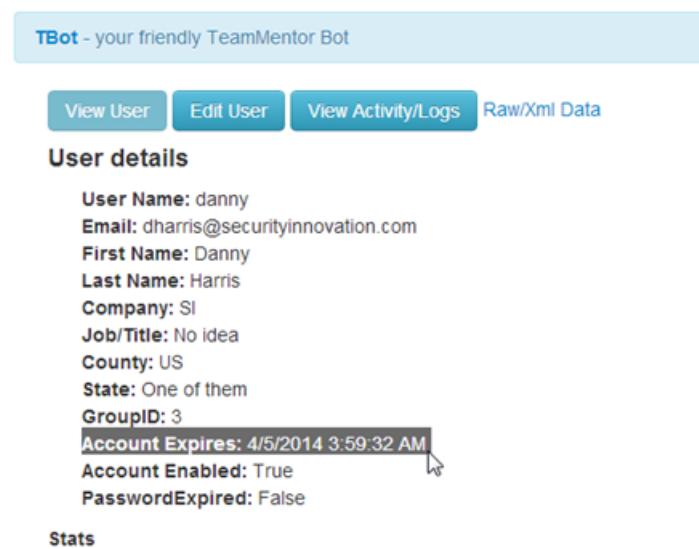
TBot - your friendly TeamMentor Bot

[View User](#) [Edit User](#) [View Activity/Logs](#) [Raw/Xml Data](#)

User details

User Name: danny
Email: dharris@securityinnovation.com
First Name: Danny
Last Name: Harris
Company: SI
Job/Title: No idea
Country: US
State: One of them
GroupId: 3
Account Expires: 4/5/2014 3:59:32 AM
Account Enabled: True
PasswordExpired: False

[Stats](#)



This is one of the nice side effects of having the ability to push TM's user data into a Git repository (another advantage is that we now have fully backed-up, logged and hashed user's change-history)

Linus gift to the world will be Git not Linux (and what about an OS built on top of an hash-driven file system?)

I know it is a big claim, but I think that [Linus Torvalds](#)⁴ will be more famous for creating Git than for this work on Linux

Linux is a great example of OpenSource development and a good OS. Its impact is mainly technical and behind the scenes.

Git is a hashed-based file system with built-in version control. Its impact is not only technical but social.

The more I use Git, the more I appreciate its beauty, simplicity and ability to scale while handling complex workflows.

See [A must watch TED talk about GIT and democracy](#)⁵ for an example of how Git can/will be used to change how information is managed in our society.

Also think about the power of having a ‘Git Powered’ OS (where all files and actions are Git controlled/tracked). We could finally get a lot of security, resilience, quality assurance and traceable from the multiple software/APIs/Apps that we use/consume.

Git also allows its technical users (like me) to be creative in finding ways to improve their productivity and workflows. See [Changing a User’s ExpiryDate from GitHub hosted file](#)⁶ or these [Git](#)⁷, [GitHub](#)⁸ and [NGit](#)⁹ posts, for examples of the wide range of areas that I have been using git for (in [TeamMentor](#)¹⁰ and [O2Platform](#)¹¹ development)

⁴http://en.wikipedia.org/wiki/Linus_Torvalds

⁵<http://blog.diniscruz.com/2012/10/a-must-watch-ted-talk-about-git-and.html>

⁶<http://blog.diniscruz.com/2013/04/changing-users-expirydate-from-github.html>

⁷<http://blog.diniscruz.com/search/label/Git>

⁸<http://blog.diniscruz.com/search/label/GitHub>

⁹<http://blog.diniscruz.com/search/label/NGit>

¹⁰<http://teammentor.net/>

¹¹<http://blog.diniscruz.com/p/owasp-o2-platform.html>

What the move from HTML to WikiText looks like (in GitHub)

Copy and paste of HTML is such a mess (even today in 2013).

I just converted a couple (converted from Word doc) TM articles from HTML into WikiText and it is shocking the difference in the amount of code (and complexity)

Example #1: Bug Database Definitions¹²

Here is what the article looks like:

The screenshot shows a web page titled "Bug Database Definitions" under the "TeamMentor" header. The header includes a logo of a person icon inside a green circle, the text "TeamMentor", and the subtitle "a Security Innovation eKnowledge Product". Below the title is a table with four columns: Technology, Category, Phase, and Type. A single row shows "TeamMentor" in the Technology column and "QA" in the Type column. Under the heading "Priority Definitions" is a section titled "Bug" with a bulleted list of priority levels: P0 (Hot customer issue), P1 (Important customer issue), P2 (Major functionality problem), P3 (Minor functionality problem), and P4 (Fit and finish). A note below states: "Customer can mean internal or external customer." Under the heading "Feature" is a bulleted list with the same five priority levels: P0 (Blocking customer adoption), P1 (Important customer request), P2 (Major functionality improvement), P3 (Minor functionality improvement), and P4 (Fit and finish). A cursor arrow is visible on the right side of the page.

Technology	Category	Phase	Type
TeamMentor			QA

Priority Definitions

Bug

- P0: Hot customer issue, fix needed within 48 hours, or application won't launch
- P1: Important customer issue, fix needed within 1 week, major functionality is completely blocked/unusable, security issue
- P2: Major functionality is experiencing a non-blocking problem or minor functionality is completely blocked
- P3: Minor functionality is experiencing a non-blocking problem
- P4: Fit and finish

Customer can mean internal or external customer.

Feature

- P0: Blocking customer adoption
- P1: Important customer request
- P2: Major functionality improvement
- P3: Minor functionality improvement
- P4: Fit and finish

And here is the 'Html to Wiki' Commit¹³

¹²<https://tm-tm4tm.azurewebsites.net/article/c649e853-99bb-45e0-8892-466ad7f4d641>

¹³https://github.com/TMContent/Lib_TM4TM/commit/ac0feb8f96b21151787a0c1f6fdf5de0806f9af4

```

https://github.com/TMContent/Lib_TM4TM/commit/ac0feb8f96b21151787a0c1f6fdf5de0806f9af4
136 - <li class="MsoNormal">Resolution
137 - note - Use this to describe why the resolution has changed.&nbsp; Eg.&nbsp;
138 - Bug is fixed in build xxx please verify and close the bug.<!--_o3a_p --></li>
139 - <li class="MsoNormal">Tech
140 - support note - Information that could be useful for tech support.<!--_o3a_p --></li>
141 - <li class="MsoNormal">Workaround
142 - -- Describe a workaround to use until the bug is fixed.</li>
143 - </ul>]]></Data>
144 + <Content Sanitized="false" DataType="wikitext">
145 + <Data><![CDATA[==Priority Definitions==
146 +====Bug===
147 +* P0: Hot customer issue, fix needed within 48 hours, or application won't launch
148 +* P1: Important customer issue, fix needed within 1 week, major functionality is completely blocked
149 +* P2: Major functionality is experiencing a non-blocking problem or minor functionality is complete
150 +* P3: Minor functionality is experiencing a non-blocking problem
151 +* P4: Fit and finish
152 +
153 +Customer can mean internal or external customer.
154 +
155 +====Feature==
156 +* P0: Blocking customer adoption

```

Example #2: Where to Post TeamMentor Issues/Comments¹⁴

 **TeamMentor**
a Security Innovation eKnowledge Product

Where to Post TeamMentor Issues/Comments

Technology	Category	Phase	Type
TeamMentor	FAQ	QA	Issue

When creating a bug, use the definitions described here: <https://tm-tm4tm.azurewebsites.net/article/c649e853-99bb-45e0>.

All new bugs should be created without a milestone set. In this way the triage team can review incoming bugs, check the labels

- For bugs, suggestions, improvements, UI issues
 - General Issues: <https://github.com/TeamMentor/Master/issues?state=open>
- Problems, suggestions for content
 - Content issues: https://github.com/TMContent/Lib_Docs/issues

And here is the ‘Html to Wiki’ Commit¹⁵

¹⁴<https://tm-tm4tm.azurewebsites.net/article/5e8f7cf1-85b2-43f4-9d15-3a560fe775ec>

¹⁵https://github.com/TMContent/Lib_TM4TM/commit/2d76a4135986a97caa4ae97645c1c5b51d031aa

https://github.com/TMContent/Lib_TM4TM/commit/2d76a4135986a97caaa4ae97645c1c5b51d031aa

```

56      -      </li>
57      -    </ul>
58      -  </ul>
59      -  <span style="font-size:12.0pt;font-family:"Times New Roman","serif""
60  -<br style="mso-special-character: line-break" /><br style="mso-special-character:line-b
61  -  <!--[if gte mso 9]><xml>
62  -  <w:WordDocument>
63  -  <w:View>Normal</w:View>
64  -  <w:Zoom>0</w:Zoom>
65  -  <w:TrackMoves></w:TrackMoves>
66  -  <w:TrackFormatting></w:TrackFormatting>
67  -  <w:PunctuationKerning></w:PunctuationKerning>
68  -  <w:ValidateAgainstSchemas></w:ValidateAgainstSchemas>
69  -  <w:SaveIfXMLInvalid>false</w:SaveIfXMLInvalid>
70  -  <w:IgnoreMixedContent>false</w:IgnoreMixedContent>
71  -  <w:AlwaysShowPlaceholderText>false</w:AlwaysShowPlaceholderText>
72  -  <w:DoNotPromoteQF></w:DoNotPromoteQF>
```

```

683  -  mso-para-margin-bottom:.0001pt;
684  -  mso-pagination:widow-orphan;
685  -  font-size:11.0pt;
686  -  font-family:"Calibri","sans-serif";
687  -  mso-ascii-font-family:Calibri;
688  -  mso-ascii-theme-font:minor-latin;
689  -  mso-hansi-font-family:Calibri;
690  -  mso-hansi-theme-font:minor-latin;}>
691  -</style>
692  -<![endif]  -->]]></Data>
17  +  <Data><![CDATA[When creating a bug, use the definitions described here: https://tm
18  +
19  +All new bugs should be created without a milestone set. In this way the triage team can
20  +** For bugs, suggestions, improvements, UI issues
21  +** General Issues: https://github.com/TeamMentor/Master/issues?state=open
22  +
23  +** Problems, suggestions for content
24  +** Content issues: https://github.com/TMContent/Lib_Docs/issues]]></Data>
693 25  </Content>
```

In this 2nd case, ironically the WikiText version looks better (look at the difference with the screenshot below with the above) because of the lack of HTML formatting mess:

 **TeamMentor**
a Security Innovation eKnowledge Product

Where to Post TeamMentor Issues/Comments

Technology	Category	Phase	Type
TeamMentor	FAQ	QA	Issue

When creating a bug, use the definitions described here: <https://tm-tm4tm.azurewebsites.net/article/c649e853-99bb-45e0-8892-466ad71>

All new bugs should be created without a milestone set. In this way the triage team can review incoming bugs, check the labels for accuracy, and prioritize them.

- For bugs, suggestions, improvements, UI issues
 - General Issues: <https://github.com/TeamMentor/Master/issues?state=open>
- Problems, suggestions for content
 - Content issues: https://github.com/TMContent/Lib_Docs/issues

Is Git a Single point of failure for TeamMentor?

Danny is getting into Git and just asked-me this:

“is it possible for Git to be a single point of failure for TM? If Git went down or offline, wouldn’t that be a problem?”

The short answer is “NO, in fact Git is a distributed point of success for TM”

Let’s start with the differences between Git and **GitHub**.

Think of Git as a ‘*file-based database of multiple versions of a particular file, with one version shown in the file system*’, i.e. ‘Git’ is the .git folder and a checkout version of the files (in the file system)

Think of GitHub as a ‘*web based location to store and share the .git folder*’

This means that a **Git** repository doesn’t go ‘*down or offline*’. A Git repository is just a .git folder, and if you wanted to remove Git **from a particular folder/repository, you could just delete that .git folder (and you would be left with the latest ‘*checked-out version of the files*’)

In terms of GitHub going down, there are two main scenarios:

1) GitHub loses the .git folders that it hosts (i.e. it loses the git repositories) - this would be a pain, but as long as there is one clone of those repositories, there shouldn’t be much/anything lost. This is why I say that ‘**Git has distributed points of success**’, basically, every clone or fork that exists, is in effect a backup of the code (there are a couple things like remote information that is not cloned, but those are minor)

2) GitHub.com is down - the impact will depend on how long this happens, at the moment we do use GitHub as a central way to distribute and publish source code between TM devs and servers, so if GitHub is down (more specifically, GitHub’s Git hosting service, not the GitHub.com website), then we can’t push the commits made (note that the devs can still work on the local git clones, and TeamMentor users can still edit Git based Libraries). Even in the case where GitHub is down for a significant time, there are easy solutions to implement (specially when you compare with the fact that if GMail, Google or Twitter goes down, there is nothing we can’t do). Part of the power of Git is that all commits are tied to its Hash, which means that while GitHub is down, we could host our own git server (see [here how I did it using apache¹⁶](#)) or push commits directly to Azure (which can also act as a Git Server). Then once GitHub is back online all we need to do is to push the commits to it.

¹⁶<http://blog.diniscruz.com/2013/03/setting-up-apache-htpd-based-git.html>

The other scenario that could happen on GitHub is that those repositories are maliciously manipulated (lets say by an internal attack that is able to add extra commits). But since of the key advantages of *Git* is that it is a DRCS ([Distributed Revision Control System¹⁷](#)) and the entire version history is present on all clones/forks, in practice, the next ‘real’ commit would fail, which should raise the alarm (i.e. that developer would need to do a git pull before it could push the code)

The way I look at *Git* is that it creates a ‘*virtual file system, fully hashed and with version control*’, which you can read more about in the post: [Linus gift to the world will be Git not Linux \(and what about an OS built on top of an hash-driven file system?\)¹⁸](#)

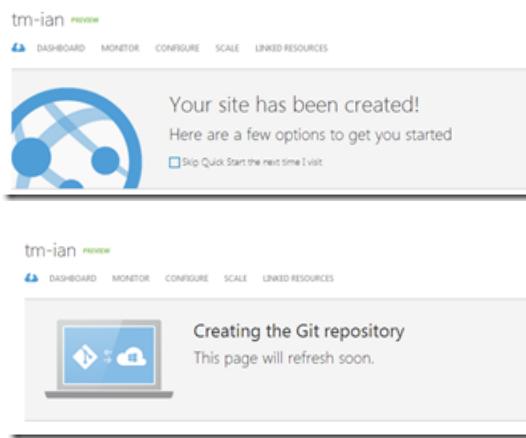
¹⁷http://en.wikipedia.org/wiki/Distributed_revision_control

¹⁸<http://www.blogger.com/blog.diniscruz.com/2013/04/linus-gift-to-world-will-be-git-not.html>

Setting up Ian's CI Development Environment (for TeamMentor)

Now that Ian (and Kofi) have pushed a couple commits (to [his fork of TeamMentor¹⁹](#)) its time to set-up Ian's CI dev environment, so that his commits can be automatically tested and viewed on a live instance of TeamMentor.

First think to do is to go to Azure and create a website to hold Ian's Fork



Which is going to be <http://tm-ian.azurewebsites.net/>²⁰

See the post [Creating a new TeamMentor test site using TeamCity, GitHub and Azure²¹](#) for more details about how TeamCity is usually configured. The only major change for Ian's version, is that TeamCity is going to track the 'azure' branch (vs the master branch)

¹⁹<https://github.com/IanIan123/Dev>

²⁰<http://tm-ian.azurewebsites.net/>

²¹<http://www.blogger.com/blog.diniscruz.com/2013/03/creating-new-teammentor-test-site-using.html>

Type of VCS	
Type of VCS:	<input type="text" value="Git"/>
VCS Root Name	
VCS Root Name:	<input type="text" value="TM Master 3.3 - Dev"/>
Enter a unique name to distinguish this VCS root from other roots. If not specified, the name will be used.	
General Settings	
Fetch URL: *	<input type="text" value="git://github.com/TeamMentor/Dev.git"/>
It is used for fetching data from repository.	
Push URL:	<input type="text"/>
It is used for pushing tags to the remote repository. If blank, the fetch url is used.	
Default Branch: *	<input type="text" value="azure"/>
Branch to be used if no branch from Branch Specification is set	

(See my next post for more details on how this azure branch was created)

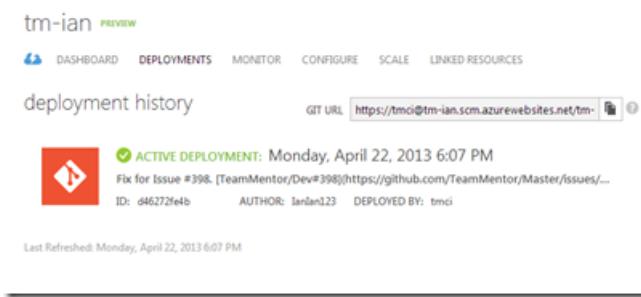
Once TM is set-up and this TeamCity build is ‘Run’ , the Azure site goes from:

The screenshot shows the Azure portal interface for a site named "tm-ian". At the top, there are navigation links: DASHBOARD, DEPLOYMENTS, MONITOR, CONFIGURE, SCALE, and LINKED RESOURCES. Below these, a large message box displays the text: "Your Git repository is ready. Windows Azure will build and deploy your web site on your next push." To the left of the text is an icon of a laptop with a cloud and a gear. At the bottom of the message box, it says "YOUR REPOSITORY".

to:

The screenshot shows the Azure portal interface for the same site "tm-ian". At the top, there are navigation links: DASHBOARD, DEPLOYMENTS, MONITOR, CONFIGURE, SCALE, and LINKED RESOURCES. Below these, a deployment log is displayed. It shows a red square icon with a white arrow pointing right, followed by the text: "DEPLOYING: Monday, April 22, 2013 6:05 PM". Underneath, it says "N/A" for ID, AUTHOR, and DEPLOYED BY. At the bottom of the log, it says "Last Refreshed: Monday, April 22, 2013 6:06 PM".

to:



And when completed we will have a clean TM site based on Ian's repository (*azure* branch):

The screenshot shows the TeamMentor application running on an Azure website. The URL is <https://tm-ian.azurewebsites.net/teamMentor>. The interface includes:

- Applied Filters:** Guidance Libraries
- Technology and Phase filters:** Two dropdown menus for filtering data.
- Table:** Shows "Showing 0 items (out of 0)" and "No data available in table".
- Selected Article Panel:**
 - Login required:** Message: "If you have a valid account, please [Login](#) if not, please sign up for an account to gain access!"
 - Sign Up Form:** Fields for Username, Password, Repeat Password, E-Mail, First Name, Last Name, Company, Title/Job, Country, and State. A "Sign Up" button is at the bottom.

Finally, so that Ian has some data to play with, I logged in as admin and quickly added a couple libraries:

The screenshot shows the TeamMentor application interface. On the left, there's a sidebar titled "Applied Filters" with a "Guidance Libraries" section containing categories like ".NET 4.0" and "Java". The main area has two filter panels: "Technology" (Any, ASP.NET 4.0) and "Phase" (Deployment, Design, Implementation, Test). Below these is a table titled "Showing 389 items (out of 389)" with columns: Title, Technology, Phase, and Type. The table lists several items, such as "A Control Flow Analysis Is Performed" (Any, Implementation, Checklist Item), "A Data Flow Analysis Is Performed" (Any, Implementation, Checklist Item), "A New ID Is Assigned on Login" (ASP.NET 4.0, Implementation, Checklist Item), and "A New Or Blank Session ID Is Assigned on Logout" (ASP.NET 4.0, Implementation, Checklist Item). To the right, a "Selected Article" panel displays the title "A Control Flow Analysis Is Performed" and sections for "What to Check For", "Why", "How To Check", and "How To Fix".

Running NUnit tests in TeamCity:

I also configured TeamCity to run the NUnit UnitTests from Ian's solution.

And in this case there is 1 test that fails:

The screenshot shows a TeamCity build summary for "TM Build For Ian (TM Dev)". The build was successful (#9, 22 Apr 13 17:13). The "Tests" tab is selected, showing one failed test: "TeamMentor.UnitTests.dll: TeamMentor.UnitTests.CoreLib (1) - Test_TM_UserData_Ex_Users_Persistance_Can_Update_User_With_Valid_Email". The details pane shows the failure message: "Tests failed: 1, passed: 123, ignored: 10".

which is also failing locally (i.e. in VisualStudio):

TeamMentor.UnitTests.CoreLib (29 tests)	1 test failed
Test_Crypto (1 test)	Success
Test_Email (2 tests)	Success
Test_HandleRequest (7 tests)	Success
Test_HandleRequest_LibraryData (1 test)	Success
Test_JavascriptCombiner (4 tests)	Success
Test_Schemas (5 tests)	Success
Test_TM_UserData (2 tests)	Success
Test_TM_UserData_Ex_Users_Persistance (2 tests)	1 test failed: One or more child tests had errors
Can_Update_User_With_Valid_Email	Failed: Expected: <System.NullReferenceException> But was: null
Cannot_Update_User_With_Invalid_Email	Success
Test_Tracking_Application (1 test)	Success

Which means that there is a side effect of one of Ian's code changes (which he will need to fix on his repo :))

Confirming Git to TeamCity to Azure

To double check that the workflow is working ok, let's make a file change:

In the TM Website Settings.js file:

```
Settings.js ×
//TM Settings
window.TM.tmVersion      = "TM 3.3";
window.TM.ArticleTitle    = "TeamMentor 3.3";

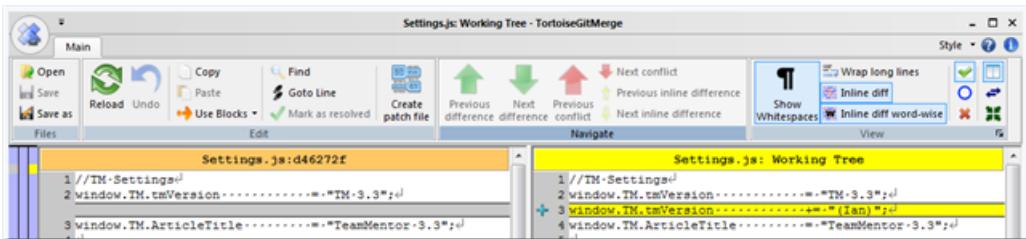
window.TM.tmWebServices   = '/Aspx_Pages/TM_WebServices.asmx/';
window.TM.NotAuthorizedPage = '/Html_Pages/Gui/Panels/AD_Non_Authorized_User.html';
```

lets append Ian's name to the build version:

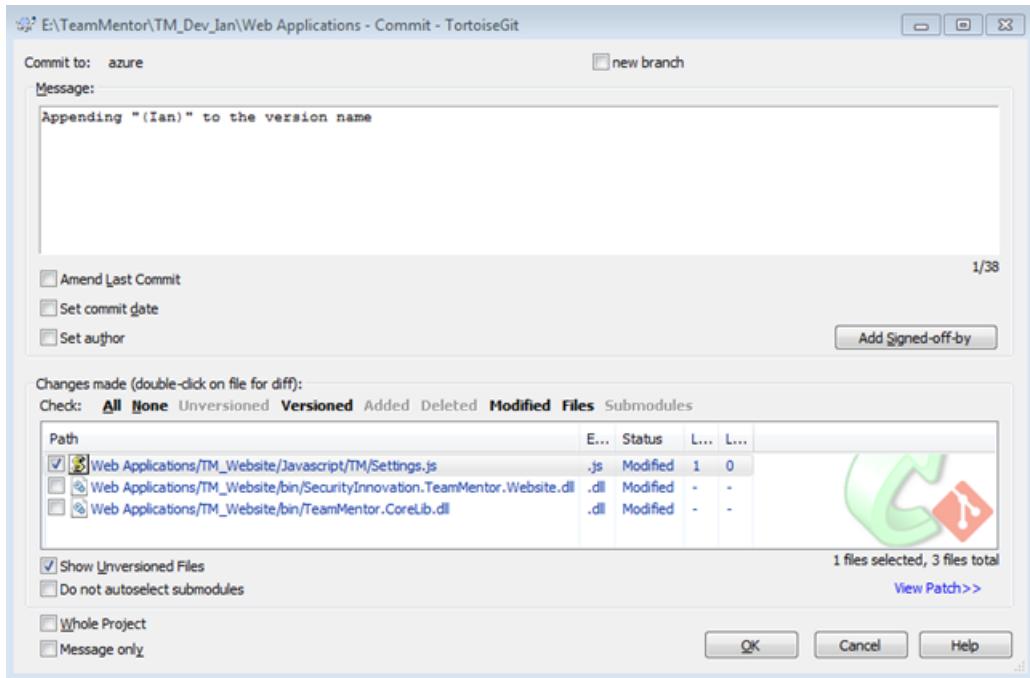
```
Settings.js ×
//TM Settings
window.TM.tmVersion      = "TM 3.3";
window.TM.tmVersion      += "(Ian)";
window.TM.ArticleTitle    = "TeamMentor 3.3";

window.TM.tmWebServices   = '/Aspx_Pages/TM_WebServices.asmx/';
window.TM.NotAuthorizedPage = '/Html_Pages/Gui/Panels/AD_Non_Authorized_User.html';
```

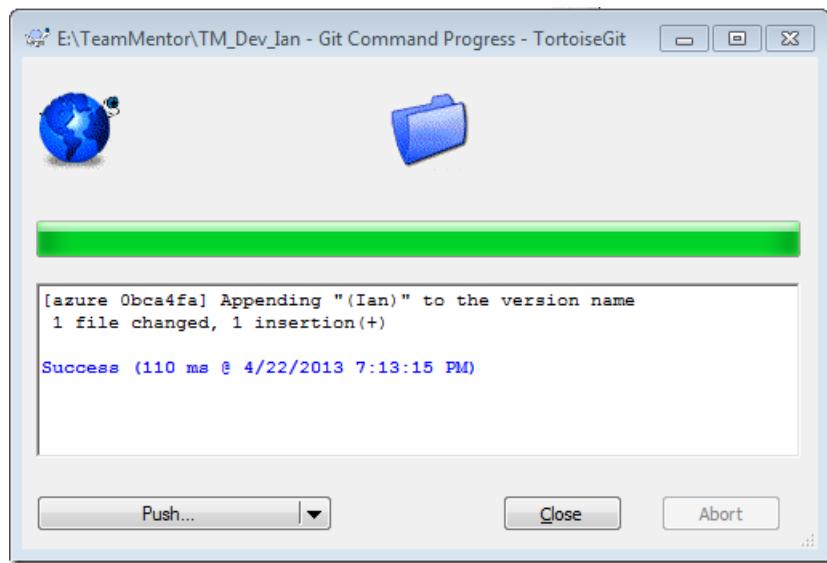
This change is picked up by Git:



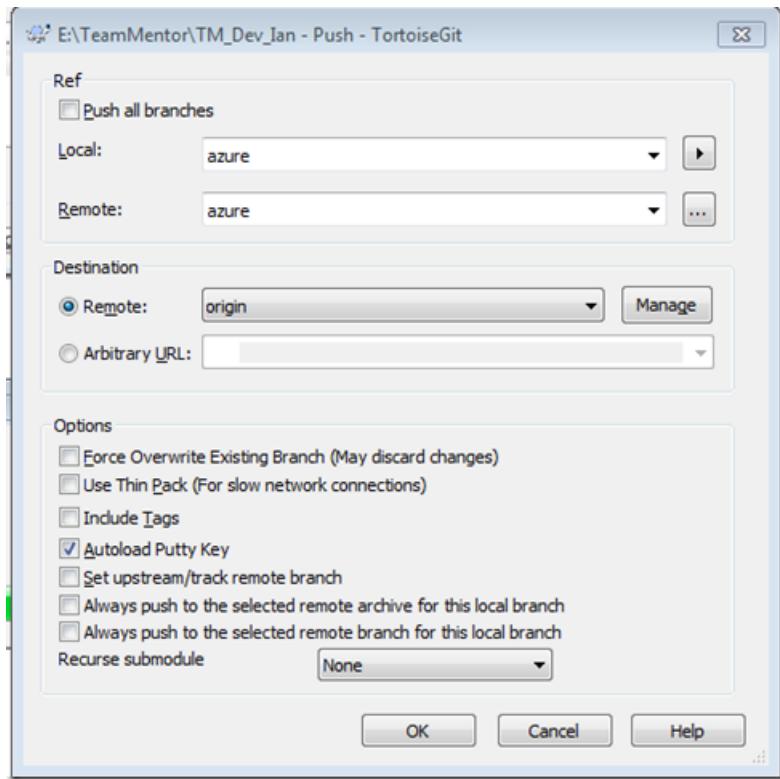
Where we can commit it locally (note the 'azure' branch)



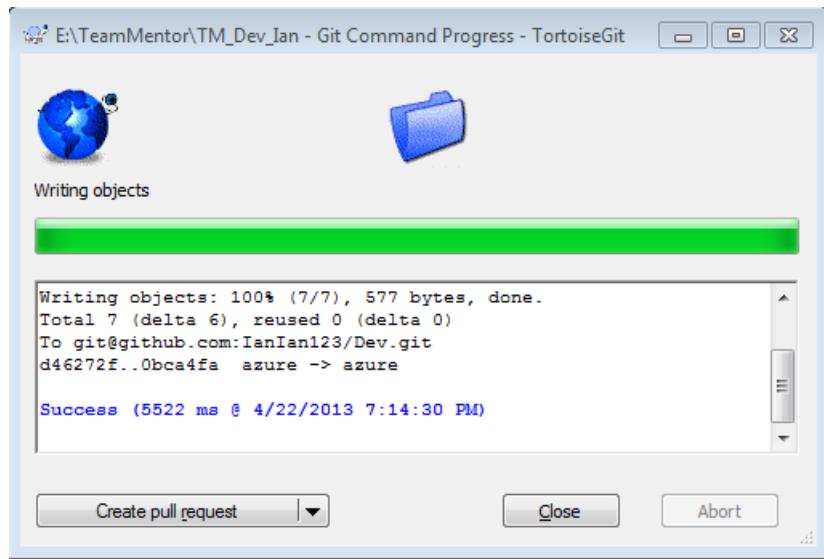
and Push it to GitHub:



Making sure both local and remote branches are ***azure***:



After the push is done:



TeamCity with trigger the build (TeamCity checks for new GitHub commits every 60sec):

A screenshot of the TeamCity web interface. The top navigation bar includes "Projects", "My Changes", "Agents", "Build Queue", "TM user", and "Administration". The main content area shows a build configuration for "TM Build For Ian (TM Dev)". It displays "Pending changes" (No pending changes) and "Current status" (1 running). A build step #10 is shown as "Running" with the ID "Obca4fa575". The status bar indicates "2m:09s left" until the build is stopped.

Azure will be pushed the new version (which is fast):

A screenshot of the Azure portal's deployment history. The top navigation bar includes "DASHBOARD", "DEPLOYMENTS", "MONITOR", "CONFIGURE", "SCALE", and "LINKED RESOURCES". The main content area shows a deployment for "deployment history" with a "GIT URL" of "https://tmci@tm-ian.scm.azurewebsites.net/tm-". The deployment status is "ACTIVE DEPLOYMENT: Monday, April 22, 2013 7:15 PM". Below this, deployment details are listed: "Appending "(ian)" to the version name", "ID: Obca4fa575", "AUTHOR: Dinis Cruz", and "DEPLOYED BY: tmci".

A refresh of the <https://tm-ian.azurewebsites.net>²² site confirms the change (notice the ‘Ian’ next to the version):

The screenshot shows a web browser window for the TeamMentor website at <https://tm-ian.azurewebsites.net/teamMentor>. The page displays a search interface with filters for 'Technology' (Any, ASP.NET 4.0) and 'Phase' (Deployment, Design, Implementation, Test). The results section shows 'Showing 389 items (out of 389)' with a 'Technology' filter applied. To the right, a 'Selected Article' panel is open, titled 'A Control Flow Analysis Is Performed'. It includes sections for 'What to Check For' (Project documentation should include the results of a control flow analysis) and 'Why' (Control flow analysis helps find vulnerabilities caused by logic errors).

Re-deploying a previous ‘Azure deployed’ version.

Note that it is also possible to go back a couple deployed versions in Azure. Just select the deployment and click on ‘REDEPLOY’

The screenshot shows the deployment history for a TeamMentor application. It lists two deployments:

- ACTIVE DEPLOYMENT:** Monday, April 22, 2013 7:22 PM
Appending "(Ian)" to the version name
ID: 0bca4fa575 AUTHOR: Dinis Cruz DEPLOYED BY: tmci
- Monday, April 22, 2013 7:20 PM**
Fix for Issue #398. [TeamMentor/Dev#398](<https://github.com/TeamMentor/Master/issues/398>)
ID: d46272fe4b AUTHOR: Ianlan123 DEPLOYED BY: tmci

At the bottom, there are 'BROWSE' and 'REDEPLOY' buttons.

Which will change the active deployment:

²²<https://tm-ian.azurewebsites.net/>

deployment history

GIT URL <https://tmci@tm-ian.scm.azurewebsites.net/tm->  

 Monday, April 22, 2013 7:15 PM
Appending "(Ian)" to the version name
ID: 0bca4fa575 AUTHOR: Dinis Cruz DEPLOYED BY: tmci

 **ACTIVE DEPLOYMENT:** Monday, April 22, 2013 7:20 PM 
Fix for Issue #398. [TeamMentor/Dev#398](<https://github.com/TeamMentor/Master/issues/398>)
ID: d46272fe4b AUTHOR: IanIan123 DEPLOYED BY: tmci

Last Refreshed: Monday, April 22, 2013 7:21 PM

7. May 2013

- Example of two TeamMentor sites using the same GitHub Content Library
- AzureGate - how Azure's 'subscription upgrade' crazy mode caused us to stop using Azure for VM Hosting (and Git+GitHub saved the day)
- Great presentation on Git Branching (very similar to the model we are using in TeamMentor)
- Fixing bug in TBot user editor via Git merge of fix developed on another repo's branch
- Releasing HotFix 1 for TeamMentor 3.3 (using Git to deploy updates to live servers)
- Great post - Git: Who cares about branches? It's all about collaboration and code reviews
- Creating o2platform.com website using GitHub Pages (with screenshots of all design options)

Example of two TeamMentor sites using the same GitHub Content Library

Now that we [TeamMentor¹](#) 3.3. is able to automatically commit, pull and push from live GitHub repositories, we are able to support quite interesting set-up and workflows.

For example at the moment there are two live TM4TM sites:

- [²](https://tm-tm4tm.azurewebsites.net) (hosted as an Azure website, deployed from TeamCity)
- [³](https://tm4tm.teammendor.net) (hosted in an Azure VM, deployed by SI's MK directly on IIS)

Both are configured to consume data from the [⁴](https://github.com/TMContent/Lib_TM4TM) repository:

¹<https://teammendor.net/>

²<https://tm-tm4tm.azurewebsites.net/>

³<https://tm4tm.teammendor.net/>

⁴https://github.com/TMContent/Lib_TM4TM

The screenshot shows a GitHub repository page for 'TMContent / Lib_TM4TM'. The repository is public. The 'Code' tab is selected. A message says 'No description or homepage.' Below it are download options: 'Clone in Windows', 'ZIP', 'HTTP', 'SSH', 'Git Read-Only', and a URL 'git@github.com:TMContent/Lib_TM4TM.git'. The 'branch' dropdown is set to 'master'. Below the branch dropdown are tabs for 'Files', 'Commits', and 'Branches' (with 1 branch). The 'Commits' section lists the following changes:

File	Time Ago	Change Description
Articles	a day ago	removed: Articles/a1c6bc9b-fd0e-4cec-baa6-180f226b5bc5.xml , A
Content Writing Process	a day ago	changed: Content Writing Process/_GuidanceItems/00000000-0000-0000-0000-000000000000
Development Process	11 days ago	changed: Development Process/_GuidanceItems/00000000-0000-0000-0000-000000000000
Guidance Articles	a day ago	changed: Guidance Articles/_GuidanceItems/00000000-0000-0000-0000-000000000000
QA	5 months ago	Added library files. [Roman87]
_Images	a month ago	Added: _Images/Contact.png , _Images/Guidance Relationships.json

Yesterday, Serge was making some changes on the [https://tm-tm4tm.azurewebsites.net⁵](https://tm-tm4tm.azurewebsites.net) server, which where automatically committed (locally) and pushed into the *Lib_TM4TM* GitHub Repo:

⁵<https://tm-tm4tm.azurewebsites.net/>

The screenshot shows a GitHub repository page for **TMContent / Lib_TM4TM**. The **Code** tab is selected. A commit history for the **master** branch is displayed, dated **Apr 30, 2013**. The commits are as follows:

- changed: Content Writing Process/_GuidanceItems/00000000-0000-0000-00...** (unknown-user authored a day ago)
- removed: Articles/a1c6bc9b-fd0e-4cec-baa6-180f226b5bc5.xml , Articles...** (unknown-user authored a day ago)
- changed: Guidance Articles/_GuidanceItems/00000000-0000-0000-0000-00...** (unknown-user authored a day ago)
- changed: Guidance Articles/_GuidanceItems/00000000-0000-0000-0000-00...** (unknown-user authored a day ago)
- changed: Guidance Articles/_GuidanceItems/00000000-0000-0000-0000-00...** (unknown-user authored a day ago)
- changed: Guidance Articles/_GuidanceItems/00000000-0000-0000-0000-00...** (unknown-user authored a day ago)

For example, here is one of the articles that Serge changed: https://tm-tm4tm.azurewebsites.net/article/Code_Example⁶

⁶https://tm-tm4tm.azurewebsites.net/article/Code_Example

Showing 1 changed file with 26 additions and 1 deletion.

[View file @ 38fd866](#)

```

27  ... @@ -26,6 +26,31 @@ also allows quickly testing from a function, inlining within other code,
28    or refactoring for a given context. The alternative is to factor up
29    front, but this increases complexity and can negatively impact
30    consumption. This leaves refactoring to the developer for their given
31    scenario. Well organized, clearly understandable code that can be cut and pasted into users project. Insightful commen
32    +scenario. Well organized, clearly understandable code that can be cut and pasted into users project. Insightful commen
33    +</li><li>Does the title distinguish it from related examples?
34    +</li><li>If technology is in the title, is the version included?
35    +</li><li>Are important nouns a user might scan for towards the left of the title? </li><ul><p><strong>Applies To:</st
36    +</li><li>Conversely, is it clear what technologies or products this does not apply to? </li><ul><p><strong>Summary:</
37    +</li><li>Is the intent of the code clear? </li><ul><p><strong>Objectives:</strong></p><ul><li>Are the user objectives
38    +</li><li>Are the objectives of the code identified?
39    +</li><li>When would a user need this code example?
40    +</li><li>Is it clear why the solution example is preferred over the problem example? </li><ul><p><strong>Scenarios:</
41    +</li><li>Do the example scenarios help highlight when to use the code or when not to? </li><ul><p><strong>Solution Ex
42    +</li><li>Do the comments add insights around decisions?
43    +</li><li>Are the comments concise enough so they don't break the code flow? </li><ul><p><strong>Problem Example:<str
44    +</li><li>Are the mistakes clear?
45    +</li><li>Are the patterns and variations of the problems clear? </li><ul><p><strong>Test Case:</strong></p><ul><li>Is
46    +</li><li>Does the example call the functional blob in the Solution example?
47    +</li><li>Can you copy+paste the code and execute it? </li><ul><p><strong>Expected Results:</strong></p><ul><li>If you
48    +</li><li>The links are directly relevant versus simply nice to have? </li><ul><p><strong>Related Items:</strong></p>
49    +</li><li>Does the title start with an action word (eg. Do something, Avoid something)?
50    +</li><li>If the item is a MUST, meaning it is prevalent and high impact, is Priority = p1?
51    +</li><li>If the item is a SHOULD, meaning it has less impact or is only applicable in narrower circumstances, is Prior
52    +</li><li>If this item will have cascading impact on application design, is Type = Design?
53    +</li><li>If this item should be followed just before deployment, is
54    +concerned with configuration details or runtime behavior, is Type =
55    +Deployment?
56    +</li><li>If this item is still in progress or not fully reviewed, is Status = Beta? </li><ul></div>]]</Data>
57    </Content>
58  </TeamMentor_Article>

```

which looks like this on the https://tm-tm4tm.azurewebsites.net/article/Code_Example⁷ server

⁷https://tm-tm4tm.azurewebsites.net/article/Code_Example

The screenshot shows a web browser window with the URL <https://tm-tm4tm.azurewebsites.net/article/00000000-0000-0000-0000-000000907236>. The page content includes:

- The links starts with the pattern "For more information on X, see ..."?
- The links are directly relevant versus simply nice to have?

Related Items:

- list of related guidance items?

Additional Tests to Consider When Writing a Test Case

Checkpoint:

- Does the title clearly state the action to take?
- Does the title start with an action word (eg. Do something, Avoid something)?
- If the item is a MUST, meaning it is prevalent and high impact, is Priority = p1?
- If the item is a SHOULD, meaning it has less impact or is only applicable in narrower circumstances, is Priority = p2?
- If the item is a COULD, meaning it is nice to know about but isn't highly prevalent or impactful, is Priority = p3?
- If this item will have cascading impact on application design, is Type = Design?
- If this item should be followed just before deployment, is concerned with configuration details or runtime behavior, is Type = Deployment?
- If this item is still in progress or not fully reviewed, is Status = Beta?

and like this on the https://tm4tm.teammentor.net/article/Code_Example⁸ server

The screenshot shows a web browser window with the URL <https://tm4tm.teammentor.net/article/00000000-0000-0000-0000-000000907236>. The page content includes:

THIS CODE EXAMPLE PROVIDES SUPPORT FOR THE FOLLOWING HOW-TO-TEST

Provide link to the How-To-Test that make use of this code example

You may find these additional articles useful

Provide links to Code Examples related to this.

Provide links to related TM Content.

Additional Resources

Provide links to outside additional resources

Specify any relevant references

Quality Check

- Provides specific technical implementation in a particular language version OR
- Provides specific configuration file exurb from a particular software
- Passes "Can I cut and paste this?" test
- Implements a difficult problem
- Is NOT something normally found in a student textbook

So at the moment the same page has different content on these servers.

⁸https://tm4tm.teammentor.net/article/Code_Example

There are two ways to fix this:

1. make a change on the <https://tm4tm.teammentor.net>⁹ server (which will trigger a git pull and push), and do a cache reset
2. do a cache reset (which will also do a git pull)

Since a cache reset will be needed on both cases, that is our best option in this case.

So, I opened the TBot page for the tm4tm.teammentor.net server:

The screenshot shows a web browser window with the URL <https://tm4tm.teammentor.net/rest/tbot/run/Commands> in the address bar. The page has a light blue header bar with the text "TBot - your friendly TeamMentor Bot". Below this, a main content area has a heading "Available TBot Commands". It contains a table with several rows of command links. The columns represent different categories: Admin, Admin\Config, Admin\LiveData, LibraryData, Messaging, and NGit. The "Admin" row includes links for "Import Legacy Users" and "Reload Server Objects" (with a cursor arrow pointing to it). The "Admin\Config" row includes "Edit GitUserLocation" and "View SecretData". The "Admin\LiveData" row includes "ActiveSessions" and "IISSessions". The "LibraryData" row includes "Articles" and "Libraries Xml". The "Messaging" row includes "Send Test Email" and "View Emails Sent". The "NGit" row includes "GitStatus".

And triggered a cache reload:

⁹<https://tm4tm.teammentor.net/>

The screenshot shows a web browser window with the URL https://tm4tm.teammentor.net/rest/tbot/run/Reload_Server_Objects. The page title is "TBot - your friendly TeamMentor Bot". Below it, the heading "Reloading Server objects" is displayed. There are two main buttons: "Reload UserData (and Git Pull and Push)" and "Reload TMConfig". A third button, "Reload Cache (will reload entire Xml database, including User data)", is highlighted with a blue border. To the right of this button, the text "Done: "In the library 'C:\TeamMentor\itm4tm\Library_Data\XmlDatabase' there are 1 library(ies), 25 views and 59 GuidanceItems"" is shown. At the bottom left, a link "back to commands list" is visible.

A quick look at the server logs confirms that a git pull took place:

The screenshot shows a web browser window with the URL <https://tm4tm.teammentor.net/rest/tbot/run/DebugInfo>. The page title is "Logs (latest first)". The log output is as follows:

```
DEBUG: [AdminAttribute] changing thread principal from System.Security.Principal.WindowsPrinc:  
DEBUG: [AdminAttribute] changing thread principal from System.Security.Principal.WindowsPrinci  
DEBUG: [API_NGIt] push completed  
INFO: [GitProgress] BeginTask : Opening connection : 0  
DEBUG: [API_NGIt] push: origin  
DEBUG: [API_NGIt] pull completed  
INFO: [GitProgress] BeginTask : Updating references : 0  
INFO: [GitProgress] BeginTask : Pull : 2  
DEBUG: [API_NGIt] open: C:\TeamMentor\itm4tm\Library_Data\XmlDatabase\TM_Libraries_TM4TM\TM4TM  
DEBUG: loaded GuidanceItems from disk in 0s:156ms  
INFO: in removeMissingGuidanceItemsIdsFromViews  
INFO: in ensureFoldersAndViewsIdsAreUnique  
DEBUG: saveGuidanceItemsToCache in 0s:15ms  
DEBUG: xmlDB_GuidanceItems in 0s:124ms
```

And the https://tm4tm.teammentor.net/article/Code_Example¹⁰ is now updated with the latest content:

¹⁰https://tm4tm.teammentor.net/article/Code_Example

The screenshot shows a web browser window with the URL https://tm4tm.teammentor.net/article/Code_Example. The page content includes:

- The links starts with the pattern "For more information on X, see ..."?
- The links are directly relevant versus simply nice to have?

Related Items:

- list of related guidance items?

Additional Tests to Consider When Writing a Test Case

Checkpoint:

- Does the title clearly state the action to take?
- Does the title start with an action word (eg. Do something, Avoid something)?
- If the item is a MUST, meaning it is prevalent and high impact, is Priority = p1?
- If the item is a SHOULD, meaning it has less impact or is only applicable in narrower circumstances, is Priority = p2?
- If the item is a COULD, meaning it is nice to know about but isn't highly prevalent or impactful, is Priority = p3?
- If this item will have cascading impact on application design, is Type = Design?
- If this item should be followed just before deployment, is concerned with configuration details or runtime behavior, is Type = Deployment?
- If this item is still in progress or not fully reviewed, is Status = Beta?

Note that the current plan is to run TBot as a constant server thread, which will then be able to monitor the GitHub's content repository and automatically do git pulls (when needed).

AzureGate - how Azure's 'subscription upgrade' crazy mode caused us to stop using Azure for VM Hosting (and Git+GitHub saved the day)

Late last night all the main TM hosted sites went down!

The reason is Azure's crazy 'subscription expired' workflow which you can read what other Azure users had to say about it when I happened to them on [http://stackoverflow.com/questions/12791020/windows-azure-virtual-machine-deleted-after-spending-limit-reached-how-can-i-g¹¹](http://stackoverflow.com/questions/12791020/windows-azure-virtual-machine-deleted-after-spending-limit-reached-how-can-i-g) and Microsoft's view on it [http://blogs.msdn.com/b/narahari/archive/2012/10/18/windows-azure-virtual-machine-disappeared-or-gone-how-do-i-recover.aspx¹²](http://blogs.msdn.com/b/narahari/archive/2012/10/18/windows-azure-virtual-machine-disappeared-or-gone-how-do-i-recover.aspx) (note how the crowd in the comments are not happy with it)

Below is the email I sent internally at SI, with my debrief on what happened:

----- (start)

Ok so we are back online with

- [https://www.teammentor.net¹³](https://www.teammentor.net)
- [https://services.teammentor.net¹⁴](https://services.teammentor.net)
- [https://owasp.teammentor.net¹⁵](https://owasp.teammentor.net)
- [https://tm4tm.teammentor.net¹⁶](https://tm4tm.teammentor.net)

Here are some notes:

- The outage lasted about 14h
- This is an example of 'worse case scenario' where our 'Data center' effectively went down
- The good parts is that:
- The prob was picked up quite quickly

¹¹<http://stackoverflow.com/questions/12791020/windows-azure-virtual-machine-deleted-after-spending-limit-reached-how-can-i-g>

¹²<http://blogs.msdn.com/b/narahari/archive/2012/10/18/windows-azure-virtual-machine-disappeared-or-gone-how-do-i-recover.aspx>

¹³<https://www.teammentor.net/>

¹⁴[http://services.teammentor.net/](https://services.teammentor.net/)

¹⁵<https://owasp.teammentor.net/>

¹⁶<https://tm4tm.teammentor.net/>

- We had a ‘fully working’ [Teammentor.net¹⁷](http://teammentor.net) site (i.e. with the content and users configured) up in about 15m (namely the one I created which was available on a direct IP)
- Roman was able to create a replacement server (for all 4 sites) in about 2hours (and that could have been 30m if Roman had not hit on a permission issue (and had done this type of deployments before))
- The fact that TM’s team spreads across multiple time-zones allowed us to react quickly
- Due to TM’s current architecture, if we did have customers who NEEDED to have access to TM guidance ASAP (see comment below) we would have had several solutions for them (including giving them a full download to run locally)
- This was the first time we really put the new 3.3. TeamMentor’s ‘auto commit and push to GitHub’ architecture in action, and I’m happy that it worked quite well (we can still fine tune it a bit, BUT this could have been MUCH worse (note that even now we still don’t have access to the old TM VMs since the VM login accounts are not working as expected, so if the userdata was not on GitHub, we would have not been able to restore the users))
- **The bad parts is that:**
- Nobody really cared :(
- Where were the ‘urgent call for TM to be up?’, twitter hashtag of TM failure!!!, ‘vulnerabilities not being fixed because TM is down’
- The sales guys were not ‘up in arms’, ‘potential sales were not affected’
- Ed or Jason were not woken up in the middle of the night with a “WTF, TM is down!!” (note: I was about to go to bed when I noticed that something was wrong in TM’s world)
- Like Roman was mentioned: “We drove for 14h on the wrong side of the road, and nothing happened”
- We found the hard way that we can’t trust Azure for live sites
- Azure really fucked us up. Roman just confirmed that the Azure GUI a couple days where still saying something like ‘your subscription is going to end in 20ish days’ (and destroying VMs and its configuration is a very crazy way to handle ‘account suspension for lack of payment or wrong subscription mode’)
- We also found a ‘single-point’ of failure where MK was the only one that could change the DNS (this is the main reason that it took 14h vs 2h/30m)
- and since ‘nobody was really complaining’ about TM’s MIA, there was no urgency to contact MK (who was dealing with a number of personal probs related to his “car losing a wheel on the motorway!”)

¹⁷<http://teammentor.net/>

Since we now have a fully working TM environment, we can spend a couple days thinking about the best place to host the TM VMS.

I like EC2, but we can go anywhere that give us good VM management

----- (end)

Great presentation on Git Branching (very similar to the model we are using in TeamMentor)

Just saw [this presentation on Git Branching¹⁸](#) (embedded below) which is really close to the model we are currently using to manage TeamMentor's app development.

I really agree with just about everything Lemi Ergin says and this is a great description of the power of Git for branching

¹⁸<http://www.slideshare.net/lemiorhan/git-branching-model>

Fixing bug in TBot user editor via Git merge of fix developed on another repo's branch

Here is an example of how I just created a HotFix branch to address an issue we want to push to our live servers asap, and how the fix was developed by Ian in one of this dev branches.

First I created _HotFix _Branch at a (freshly baked) local clone of the ***TeamMentor/Dev*** repository:

```
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (master)
$ git checkout -b HotFix_3_3_1
```

Then I reviewed the code from Ian's branch I want to merge:

The screenshot shows a GitHub comparison view between the master branch and a pull request (#437-Password-Expiry). The commit message is "Date value being saved to database." The diff shows the following changes in the `User_Edit.cshtml` file:

```

@@ -5,9 +5,12 @@
    $scope.update = function(userData)
    {
        $scope.result = "... saving user";
+
+       userData.ExpirationDate = "/Date(" + Date.parse(userData.date) + ")\\/";
+       alert('d: ' + userData.ExpirationDate);
+       $http.put('/rest/user',userData)
+           .success(function(data)
+           {
+               $scope.result = (data) ? "(User details saved)" : "Error: Failed to Save user"
+           })
+
@@ -84,8 +87,8 @@
    </div><br/>-->
    <div class="input-prepend">
        <span class="add-on" style="width:120px;">Account Expiration</span>
        <input type="text" style="width:250px;" ng-model="userData.date"/>
    </div> (read only)<br/>
    <input type="text" style="width:250px;" ng-model="userData.date" required/>
</div><br/>
<label class="checkbox">
```

When happy with the changes, I used a *git fetch** to get the latest version of Ian's fork of **_TeamMentor/Dev*

```
02@WIN-FGNQ5AARJ80 /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (HotFix_3_3_1)
$ git fetch Dev_Ian
```

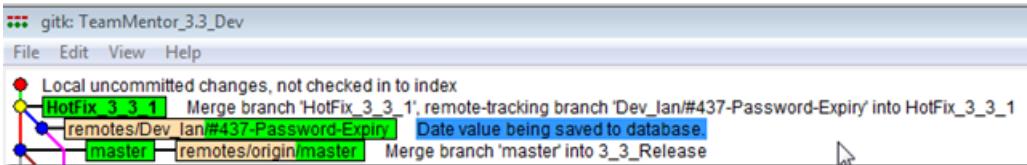
```
remote: Counting objects: 293, done.
remote: Compressing objects: 100% (116/116), done.
Receiving objects: 62% (143/230), 76.00 KiB | 57 KiB
Receiving objects: 100% (230/230), 132.45 KiB | 57 KiB/s, done.
Resolving deltas: 100% (192/192), completed with 56 local objects.
From github.com:IanIan123/Dev
 * [new branch]      #170-My-account-section-allows-to-save-insecure-password ->
to-save-insecure-password
 * [new branch]      #352          -> Dev_Ian/#352
 * [new branch]      #353          -> Dev_Ian/#353
 * [new branch]      #413          -> Dev_Ian/#413
 * [new branch]      #437-Password-Expiry -> Dev_Ian/#437-Password-Expiry
 * [new branch]      #445_Article_Preview -> Dev_Ian/#445_Article_Preview
 * [new branch]      NodeJS        -> Dev_Ian/NodeJS
 * [new branch]      azure         -> Dev_Ian/azure
 * [new branch]      master         -> Dev_Ian/master ...
```

Followed by a `$ git merge Dev_Ian/#437-Password-Expiry HotFix_3_3_1`

```
$ git merge Dev_Ian/#437-Password-Expiry HotFix_3_3_1
Trying simple merge with Dev_Ian/#437-Password-Expiry
Already up-to-date with HotFix_3_3_1
Merge made by the 'octopus' strategy.
 Web Applications/TM_Website/TBot/Views/User_Edit.cshtml      | 9 ++++++---
 .../TM_AppCode/Users/TM_UserData_Ex_Users.cs                  | 4 +---
 .../TM_AppCode/Users/TM_UserData_Ex_Users_Persistance.cs    | 3 +-+
 .../TM_AppCode/WebServices/TM_WebServices.Users.cs           | 4 +---
 .../TM_AppCode/WebServices_REST/Controllers/TM_REST_Users.cs | 4 +--+
5 files changed, 14 insertions(+), 10 deletions(-)
```

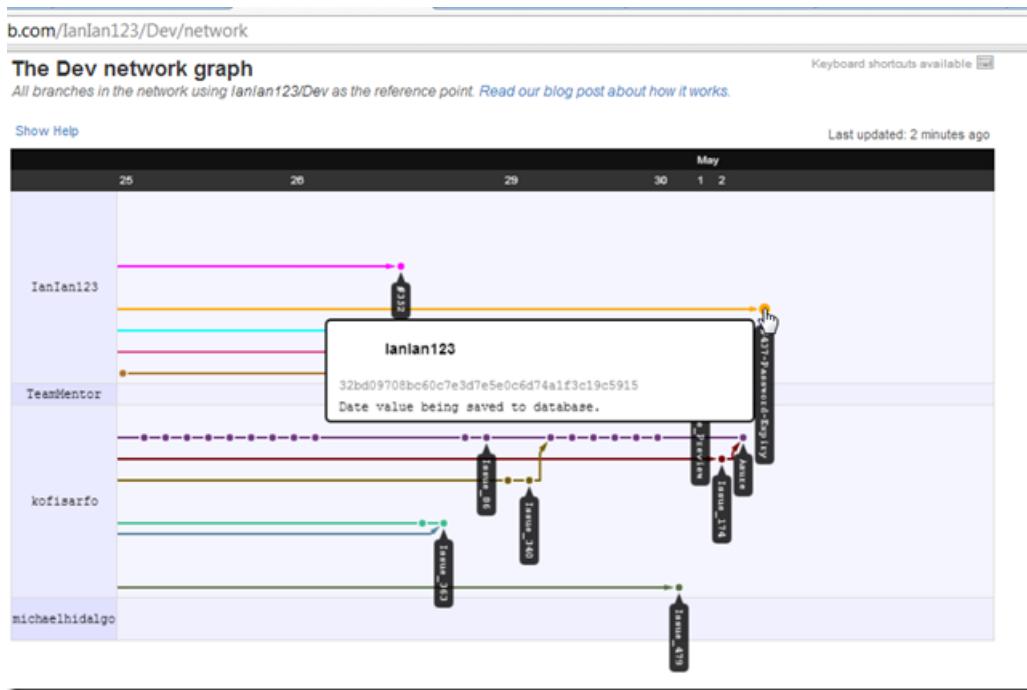
Which did the merge with the _ TeamMentor/Dev_ master branch

A look at GitK confirms that there was only one commit added (the Ian's 'Date value being saved to database'¹⁹)



At this stage if we look at Ian's network map, we will see that this commit is not linked to another commit (i.e. is the last one of the **#437-Password-Expiry** branch

¹⁹<https://github.com/IanIan123/Dev/commit/32bd09708bc60c7e3d7e5e0c6d74a1f3c19c5915>



Next step is to quickly test if the feature is working ok.

This fix is for the [Password expiry cannot be set from the main TM GUI²⁰](#) issue (i.e. make the 'Account Expiration' field editable).

So I opened an user's edit page

²⁰<https://github.com/TeamMentor/Master/issues/437>

TBot - your friendly TeamMentor Bot

View User Edit User View Activity/Logs Raw/XML Data

Editing user

User ID 1206801006

UserName	test
Email	khavimlcwe@randomm.xyz
FirstName	a
LastName	test
Company	...
Job/Title	...
Country	...
State	...
Account Expiration	Sun Jan 05 2014 00:00:00 GMT+0000 (

Password Expired
 User Enabled

Save back to user's list

changed the expiration date:

localhost:3187/rest/tbot/run/User_Edit?test

TBot - your friendly TeamMentor Bot

View User Edit User View Activity/Logs Raw/XML Data

Editing user

User ID 1206801006

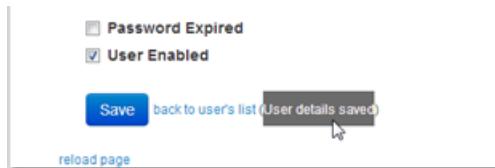
UserName	test
Email	khavimlcwe@randomm.xyz
FirstName	a
LastName	test
Company	...
Job/Title	...
Country	...
State	...

Account Expiration Sun Jul 20 2014 00:00:00 GMT+0000 (CET)

Password Expired
 User Enabled

Save back to user's list

Saved it



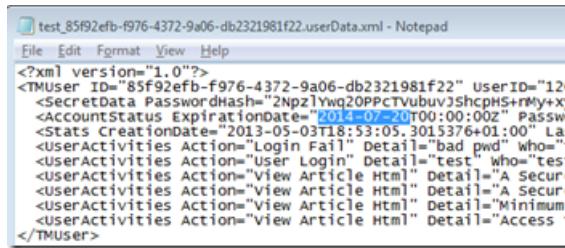
And confirmed that the user's xml data was changed on the in-memory version of the user xml files:

User details

Raw Data

```
<?xml version="1.0"?>
<TMUser ID="85f92efb-f976-4372-9a06-db2321981f22" UserID="120680
ate="" GroupID="1">
<SecretData PasswordHash="2Npz1Ywq20PPcTVubuvJShcpHS+nMy+xyTum
" />
<AccountStatus ExpirationDate="2014-07-20T00:00:00Z" PasswordE
<Stats CreationDate="2013-05-03T18:53:05.3015376+01:00" LastLo
```

and on the file system:



My final step was to push the HotFix branch into the live server:

```
c2@WIN-FGNQ5AARJ8O /e/TeamMentor/_TM_DEV/TeamMentor_3.3_Dev (HotFix_3_3_1)
$ git push origin HotFix_3_3_1:HotFix_3_3_1
Counting objects: 53, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (31/31), 35.76 KiB, done.
Total 31 (delta 28), reused 17 (delta 16)
To git@github.com:TeamMentor/Dev.git
 * [new branch]      HotFix_3_3_1 -> HotFix_3_3_1
```

Here are the commits in the new ***HotFix_3_3_1*** branch (note the Ian's ‘[Date value being saved to database²¹](#)’ branch is now there)

²¹<https://github.com/IanIan123/Dev/commit/32bd09708bc60c7e3d7e5e0c6d74a1f3c19c5915>

branch: HotFix_3_3_1

Files Commits Branches Tags

Dev / Commit History

Keyboard shortcuts available

May 03, 2013

- Removing Javascript alert from save submission
DinisCruz authored 8 minutes ago [9ed1cabd73](#) [Browse code](#)
- Merge branch 'HotFix_3_3_1', remote-tracking branch 'Dev_Ian/#437-Pas...'
DinisCruz authored 25 minutes ago [085231f960](#) [Browse code](#)

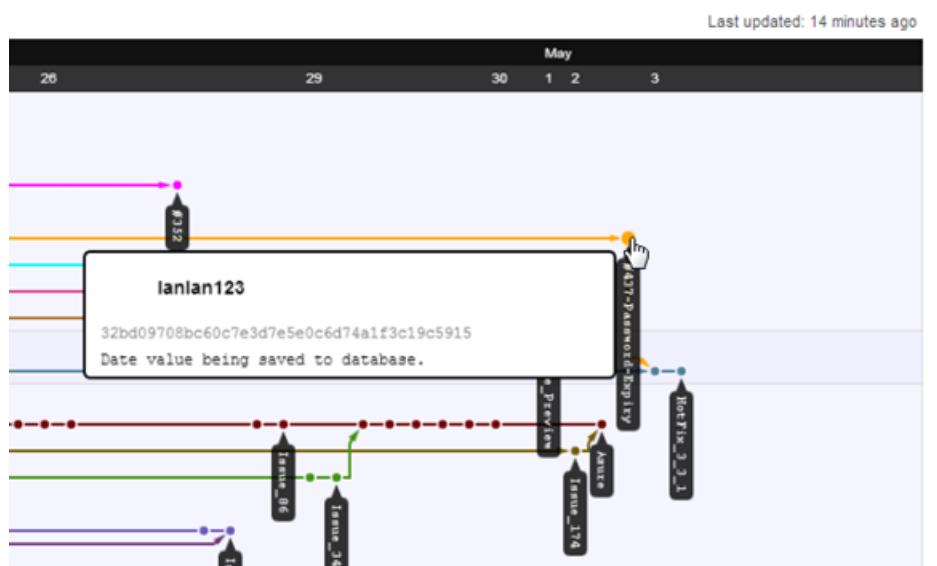
May 02, 2013

- Date value being saved to database.
ianlan123 authored 19 hours ago [32bd09705b](#) [Browse code](#)

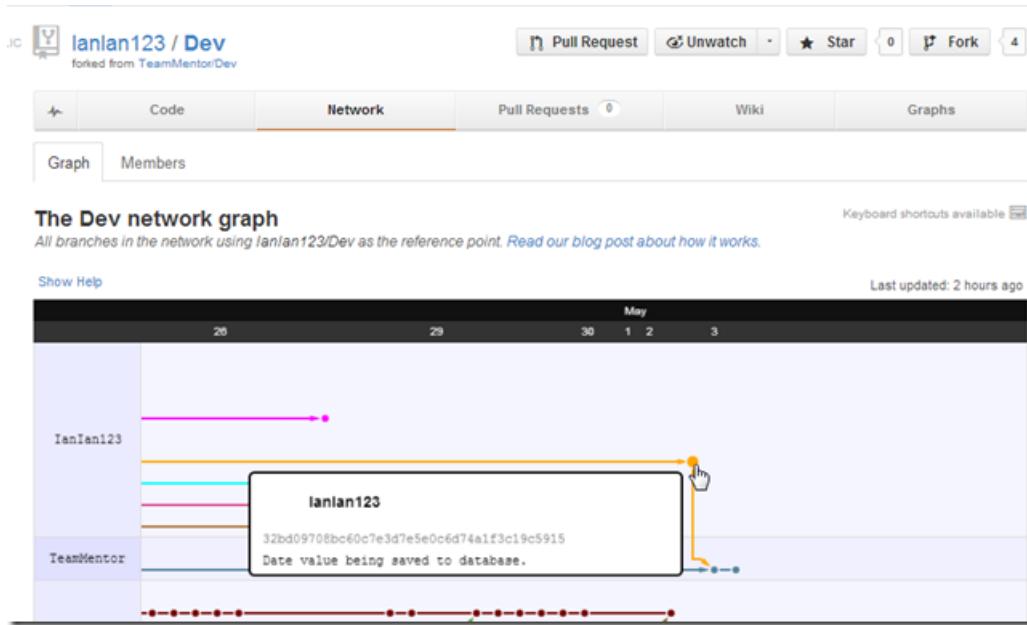
Apr 15, 2013

- Merge branch 'master' into 3_3_Release
DinisCruz authored 18 days ago [e004e1d3cb](#) [Browse code](#)

And now Ian's branch is connected with the new HotFix_3_3_1 branch:



same graph without the branch labels:



Releasing HotFix 1 for TeamMentor 3.3 (using Git to deploy updates to live servers)

This is how I updated the 3.3 version of TeamMentor to 3.3.1, which contained a fix for the [Password expiry cannot be set from the main TM GUI²²](#) issue

Since we are now using the [Vincent Driessen GitFlow branching model²³](#) (see also these [Git-Flow²⁴](#) scripts and this [great presentation²⁵](#)), after the [issue 437²⁶](#) was reported+prioritised, all development happened on a Feature Branch called [HotFix_3_3_1²⁷](#) (which was created from a Pull Request from Ian's own [437-Password-Expiry²⁸](#) dev branch).

Once TM's QA (ie. Roman) was happy with the patch, it was time to push it to the first batch of TM production servers (my responsibility was to update the [sme.teammentor.net²⁹](#) server, while Roman updated the [tm4tm.teammentor.net³⁰](#), and the rest will be updated by Michael K + Michael H). Technically I did a pull from those servers :)

Here are my steps:

- 1) I RDP into the production server and opened up the respective TM folder (which is a clone of <https://github.com/TeamMentor/Master>)

²²<https://github.com/TeamMentor/Master/issues/437>

²³<http://nvie.com/posts/a-successful-git-branching-model/>

²⁴<http://yakiloo.com/getting-started-git-flow/>

²⁵<http://blog.diniscruz.com/2013/05/great-presentation-on-git-branching.html>

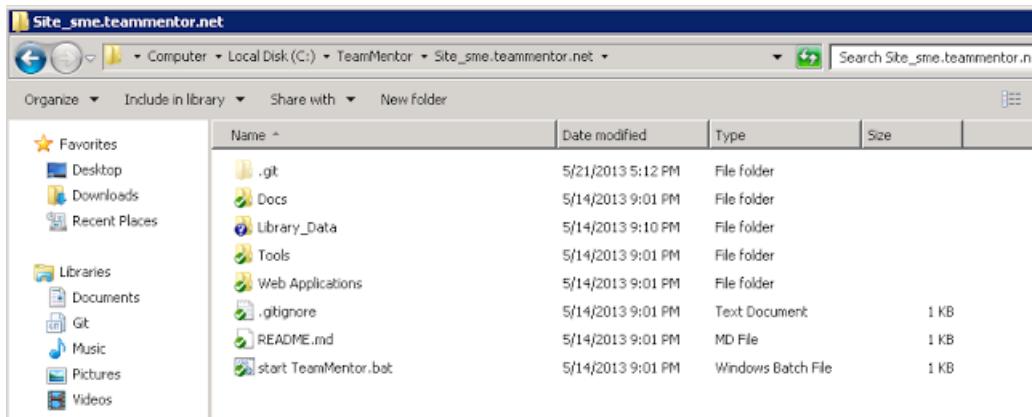
²⁶<https://github.com/TeamMentor/Master/issues/437>

²⁷https://github.com/TeamMentor/Dev/tree/HotFix_3_3_1

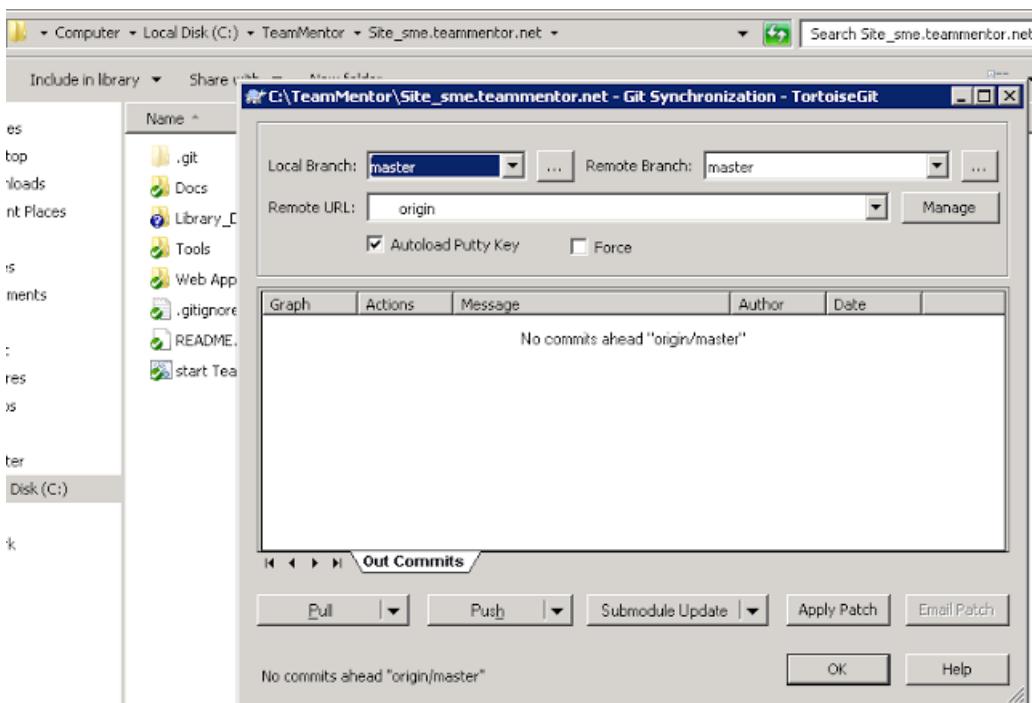
²⁸<https://github.com/IanLan123/Dev/commits/%23437-Password-Expiry>

²⁹<http://sme.teammentor.net/>

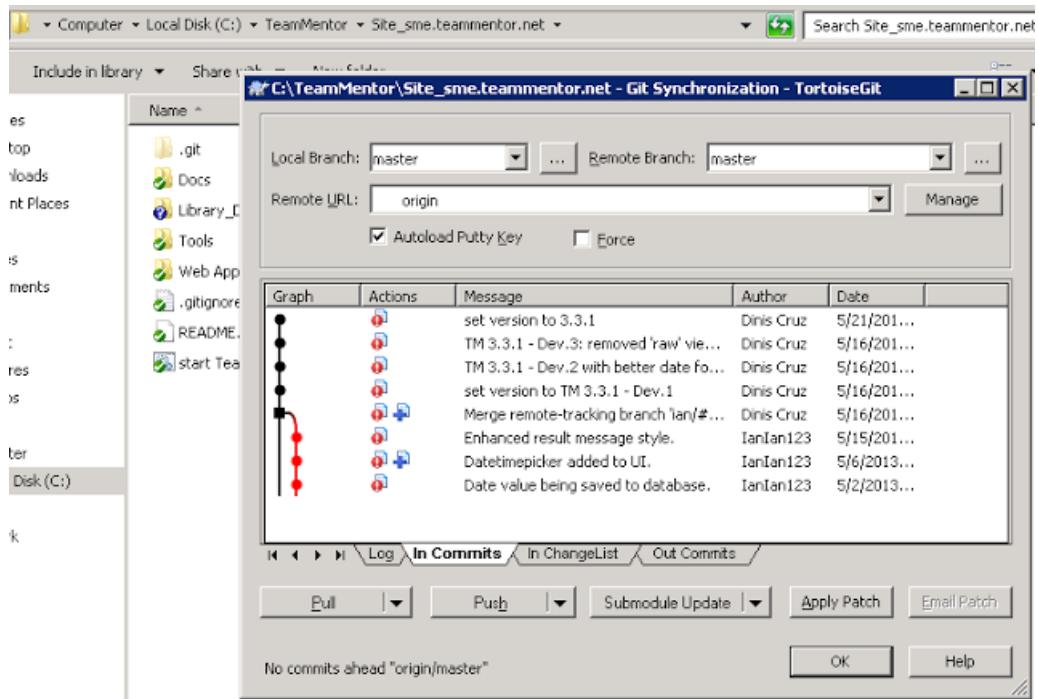
³⁰<http://tm4tm.teammentor.net/>



2) opened up the Git Sync gui



3) and did a Pull



And that's it, update done :)

I would say that this took me about 1m and Roman says he did his in 58 secs :)

Just to make sure all is good:

4) opened up sme.teammentor.net (note the new version number on top right)

The screenshot shows the TeamMentor web application interface. At the top, there is a header bar with a back arrow, forward arrow, refresh button, and a URL field showing <https://sme.teammentor.net/teamMentor>. To the right of the URL are icons for star, three dots, and a search/magnifying glass.

The main content area has a blue header bar with the TeamMentor logo and the text "a Security Innovation eKnowledge Product". On the right side of the header, there are links for "Edit Mode", "Change Password", "Control Panel", "Logout", "TM 3.3.1", and "About Help". Below the header, it says "Logged in as: dlinis".

The interface is divided into several sections:

- Applied Filters:** A sidebar on the left containing a "Guidance Libraries" button.
- Search:** A search bar at the top center.
- Technology and Phase:** A central grid area with two columns labeled "Technology" and "Phase". Both columns are currently empty.
- Selected Article:** A sidebar on the right.

5) and confirmed that all was good (first load took a couple secs since there was a server side IIS W3WP process refresh, caused by the updated bin folder)

The screenshot shows a web browser window for <https://sme.teammendor.net/teamMentor>. The page title is "TeamMentor" with the subtitle "a Security Innovation eKnowledge Product". The top right includes links for "Edit Mode", "Change Password", "Control Panel", "Logout", and "TM 3.3.1". A user is logged in as "dinis". The main content area has tabs for "Applied Filters" and "Selected Article". The "Applied Filters" tab is active, showing filters for "Technology" (Any, .NET 2.0, .NET 2.0, ADO.NET 2.0, ASP.NET 2.0, Any, SQL Server 2000, Web Application) and "Phase" (Deployment, Design, Implementation, Test). Below these filters, a message says "Loaded 1454 out of 1454". The "Selected Article" tab shows a single article titled "A Centralized Log Server Is Deployed". The article details are: Title "A Centralized Log Server Is Deployed", Technology "Any", Phase "Deployment". Below this, there are four more article snippets: "A Certificate Is Installed on the Database Server to Support SSL Communication", "A Control Flow Analysis Is Performed", "A Custom ASP.NET Policy Is Used to Access Non-SQL Server Databases from Partial Trust ASP.NET Applications.", and "A Custom Least-privileged Anonymous Account Is Created for Anonymous Access." Each snippet lists its Technology (SQL Server 2000, Any, ADO.NET 2.0, Web Application) and Phase (Deployment, Implementation, Deployment).

6) I then opened TBot's user management page (for my user)

TBot - your friendly TeamMentor Bot

[View User](#) [Edit User](#) [View Activity/Logs](#) [Raw/XML Data](#)

Editing user

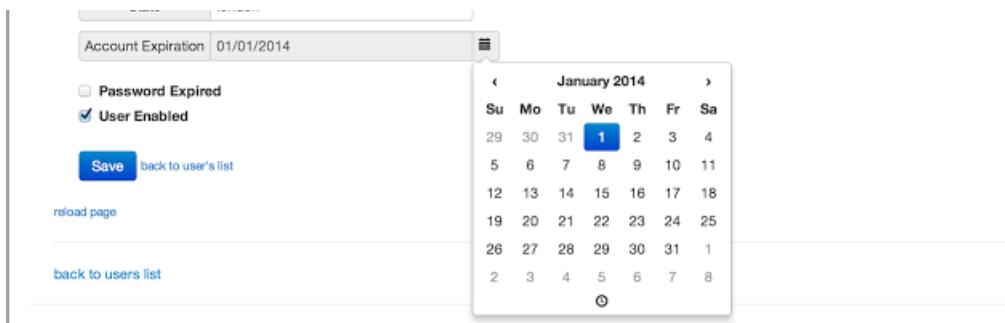
User ID 149893989

UserName	dinis
Email	dcruz@securityinnovation.com
FirstName	dinis
LastName	cruz
Company	Si
Job/Title	dev
Country	UK
State	london
Account Expiration	01/01/1901

Password Expired
 User Enabled

[Save](#)

7) changed the expiration date (this was the issue 43³¹ since it was not possible to change this value from the web (only via GitHub (see [Changing a User's ExpiryDate from GitHub hosted file³²](#)) or REPL script (see [Using CSharpRepl to batch change TeamMentor's users email and settings³³](#)))



8) Saved the changes

³¹<https://github.com/TeamMentor/Master/issues/437>

³²<http://blog.diniscruz.com/2013/04/changing-users-expirydate-from-github.html>

³³<http://blog.diniscruz.com/2013/04/using-csharprepr-to-batch-change.html>

Account Expiration: 01/01/2014

Password Expired
 User Enabled

Save back to user's list (User details saved)

[reload page](#)

9) and confirmed that they were committed ok locally (and into GitHub)

The screenshot shows a GitHub commit page for a repository named `TMClients / Site_sme.teammentor.net`. The commit is titled `changed: Users/dinis_f168b566-6676-4feb-b5b2-396b786c5aac.userData.xml`. It was authored by an unknown user just now. The commit message indicates it has 1 parent (`8ca364b`) and a commit hash of `b9db43b6e27ae6e`. The commit shows 1 changed file with 1 addition and 1 deletion. The diff view highlights changes in the XML file:

```

diff --git a/Users/dinis_f168b566-6676-4feb-b5b2-396b786c5aac.userData.xml b/Users/dinis_f168b566-6676-4feb-b5b2-396b786c5aac.userData.xml
--- a/Users/dinis_f168b566-6676-4feb-b5b2-396b786c5aac.userData.xml
+++ b/Users/dinis_f168b566-6676-4feb-b5b2-396b786c5aac.userData.xml
@@ -1,7 +1,7 @@
<?xml version="1.0"?>
<TMUser ID="f168b566-6676-4feb-b5b2-396b786c5aac" UserID="149893989" UserName=""
<SecretData PasswordHash="4GXNK2YvrGmEabryndRcWjKeNGgfpF+y2rC+lfe/tJbdIVcaQaq
- <AccountStatus ExpirationDate="0001-01-01T00:00:00" PasswordExpired="false" U
+ <AccountStatus ExpirationDate="2014-01-01T00:00:00Z" PasswordExpired="false"
<Stats CreationDate="2013-05-14T21:16:57.1836872+00:00" LastLogin="2013-05-16
<UserActivities Action="User Login" Detail="dinus" Who="dinus" IPAddress="82.
<UserActivities Action="View Article Html" Detail="A Centralized Log Server I

```

0 notes on commit `b9db43b`

GitHub's Graphs and Git helper pages

One of the great things about using GitHub as part of our workflow is that we can use its Graphing capabilities to visualize what is happening.

For example

Here is the last commit:

```

1 //TM Settings
2 window.TM.tmVersion      = "TM 3.3.1";
3 window.TM.ArticleTitle    = "TeamMentor 3.3";
4
5 window.TM.tmWebServices   = '/Aspx_Pages/TM_WebServices.asmx/';
6 window.TM.NotAuthorizedPage = '/Html_Pages/Gui/Panels/AD_Non_Authorized_User.html';
7

```

Which just changes the version number

May 21, 2013

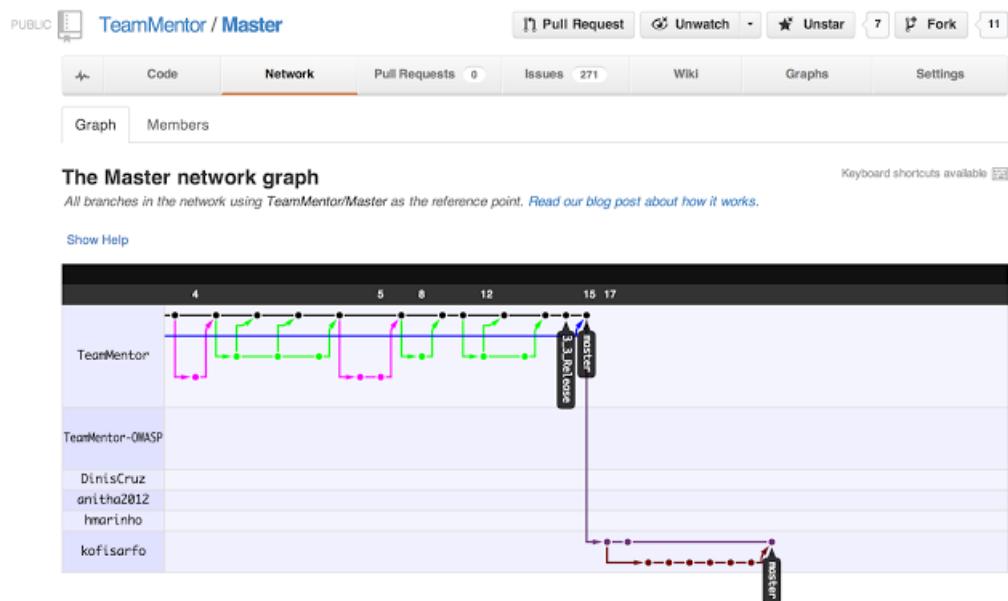
set version to 3.3.1
DinisCruz authored 3 hours ago [d9eb1677a2](#) [Browse code](#)

May 16, 2013

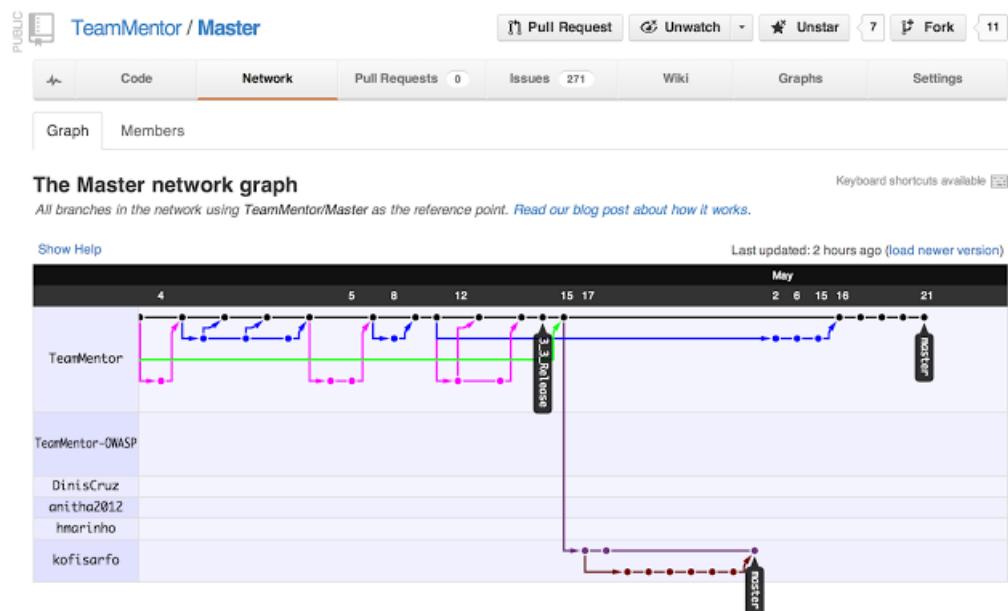
- TM 3.3.1 - Dev.3: removed 'raw' view from User_Edit.cshtml**
DinisCruz authored 5 days ago [d0085dc2d2](#) [Browse code](#)
- TM 3.3.1 - Dev.2 with better date formating on multiple users edit an...**
DinisCruz authored 5 days ago [b25222796f](#) [Browse code](#)
- set version to TM 3.3.1 - Dev.1**
DinisCruz authored 5 days ago [544342c6f7](#) [Browse code](#)
- Merge remote-tracking branch 'ian/#437-Password-Expiry' into 437-Pass...**
DinisCruz authored 5 days ago [13cf950ed0](#) [Browse code](#)

Note that this commit happen on the HotFix branch. So the next step is to push it into TeamMentor/Master (the production repo)

This is what TeamMentor/Master looks like for 3.3 (before hotfix commits)

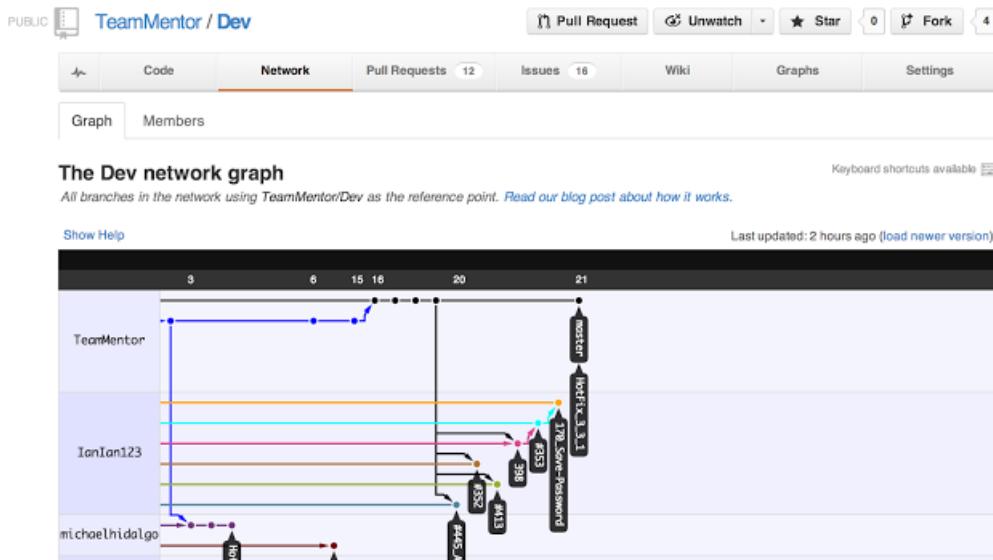


This is what TeamMentor/Master looks like after the 3.3.1 commit



And this is what the TeamMentor/Dev Repo looks like after the commit (this is the

development clone/fork, which has all the other feature branches)



I also update the Tags (using the commands described here [Adding Tags to TeamMentor Master repository³⁴](#))

Here are the git tags before the push:

³⁴<http://blog.diniscruz.com/2012/10/adding-tags-to-teammentor-master.html>

Tags

a month ago	v3.3 - 3.3 Release
	↳ e004e1d [zip] [tar.gz]
6 months ago	v3.2.4 - 3.2.4 Release
	↳ 4e39135 [zip] [tar.gz]
7 months ago	v3.2 - 3.2 Release
	↳ 4a0e621 [zip] [tar.gz]
7 months ago	v3.1 - 3.1 Release
	↳ f71b016 [zip] [tar.gz]
7 months ago	v3.0 - 3.0 Release
	↳ d9e12da [zip] [tar.gz]

** Here are the Git tags after the tag push**

Tags

2 hours ago	v3.3.1 - 3.3 HotFix 1
	↳ c9eb167 [zip] [tar.gz]
a month ago	v3.3 - 3.3 Release
	↳ e004e1d [zip] [tar.gz]
6 months ago	v3.2.4 - 3.2.4 Release
	↳ 4e39135 [zip] [tar.gz]
7 months ago	v3.2 - 3.2 Release
	↳ 4a0e621 [zip] [tar.gz]
7 months ago	v3.1 - 3.1 Release
	↳ f71b016 [zip] [tar.gz]
7 months ago	v3.0 - 3.0 Release
	↳ d9e12da [zip] [tar.gz]

Great post - Git: Who cares about branches? It's all about collaboration and code reviews

This is a great explanation of the power of Git : [Who cares about branches? It's all about collaboration and code reviews³⁵](#)

here is the author's *TL;DR*:

TL;DR: Using Git has made our team much better by removing barriers to collaboration and code reviews. Those are the real Git benefits, not specific features like fast branches.

I completely agree and it is the ability to easily review code (and send code back for rewrite during pull requests) that really makes git powerful :)

On this topic also read [Great presentation on Git Branching \(very similar to the model we are using in TeamMentor\)](#)³⁶

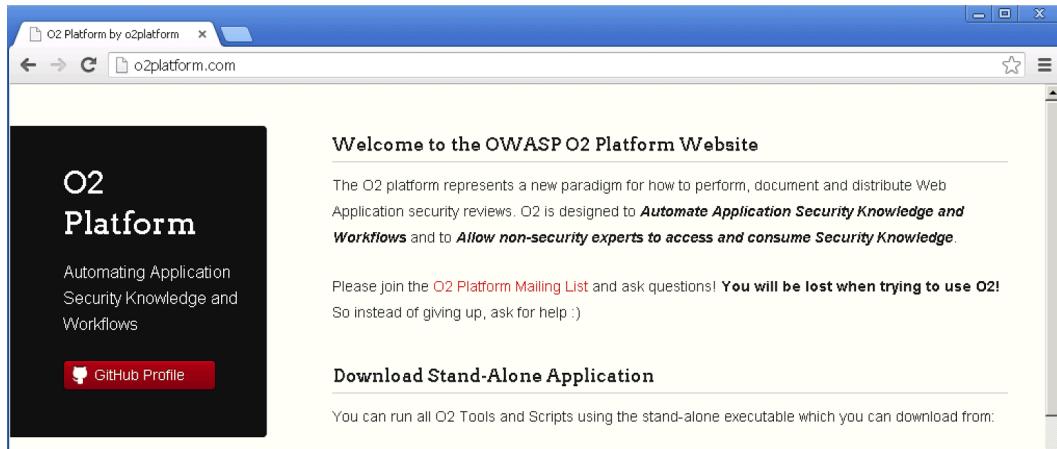
³⁵<http://blog.8thcolor.com/2013/05/git-who-care-about-branches-its-all-about-collaboration-and-code-reviews/>

³⁶<http://blog.diniscruz.com/2013/05/great-presentation-on-git-branching.html>

Creating o2platform.com website using GitHub Pages (with screenshots of all design options)

In order to set-up a site for the [http://o2platform.com³⁷](http://o2platform.com) domain, I just used GitHub Pages to create and publish a brand new site.

This is what it looks like:



What do you think?

Please see below the other design options and let me know if I made a mistake. Also if you want help in editing this site, ping me with your GitHub account, and I'll give you push access (or fork the [o2platform.github.com³⁸](https://github.com/o2platform/o2platform.github.com) repo and send me pull requests)

Here is how I created it:

I went to the [https://github.com/o2platform/o2platform.github.com^{39**}](https://github.com/o2platform/o2platform.github.com) repo, and on the settings page, I clicked on the **Automatic Page Generator

³⁷<http://o2platform.com/>

³⁸<https://github.com/o2platform/o2platform.github.com>

³⁹<https://github.com/o2platform/o2platform.github.com>

The screenshot shows the GitHub Pages interface. At the top, it says "GitHub Pages". Below that, it states "Your site is published at <http://o2platform.com>". There are two main sections: "Update your site" and "Automatic Page Generator". Under "Update your site", there is a note: "Easily change your content or theme with the page generator." Another note below it says: "To publish a page manually, push an HTML or [jekyll](#) site to the `master` branch. [More info.](#)"

which opened up this page, where I added the content that is current on this blog's [O2 Platform](#)⁴⁰ page:

The screenshot shows the GitHub user page creation interface for the repository [o2platform / o2platform.github.com](#). The title is "Create a GitHub User Page". It has fields for "Page Name" (set to "O2platform.github.com"), "Tagline" (set to "Automating Application Security Knowledge and Workflows"), and "Body (parsed with GitHub Flavored Markdown)". The body contains the following content:

```

### Welcome to the OWASP O2 Platform Website

The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to _**Automate Application Security Knowledge and Workflows**_ and to _**Allow non-security experts to access and consume Security Knowledge**_.

Please join the [O2 Platform Mailing List](https://groups.google.com/a/owasp.org/d/forum/o2-platform) and ask questions! **You will be lost when trying to use O2!** So instead of giving up, ask for help :)

### Download Stand-Alone Application

You can run all O2 Tools and Scripts using the stand-alone executable which you can download from:

```

When finished I clicked on *Continue to Layouts*

⁴⁰<http://blog.diniscruz.com/p/owasp-o2-platform.html>

```
### For more info on O2 see:  
* O2 related posts on Dinis Cruz blog (194 at last count): http://diniscruz.blogspot.co.uk/search/label/O2  
Platform  
* O2 Blog: https://o2platform.wordpress.com  
* [What is the O2 Platform](https://speakerdeck.com/diniscruz/owasp-o2-platform-automating-security-  
knowledge-through-unit-tests) presentation at SpeakerDeck
```

Google Analytics Tracking ID

UA-16441316-1

Cancel **Continue to Layouts**

And I checkout the multiple design options:

Slate:

The screenshot shows the Slides app interface with five design options displayed: "Slate" (selected), "MERLOT", "Time Machine", "MINIMAL", and "Leap Day". Below the designs, there are navigation icons for "HIDE", "EDIT", and "PUBLISH", and a "View on GitHub" button.

O2platform.github.com

Automating Application Security Knowledge and Workflows

Welcome to the OWASP O2 Platform Website

The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to **Automate Application Security Knowledge and Workflows** and to **Allow non-security experts to access and consume Security Knowledge**.

Merlot:**

The screenshot shows a web-based page builder or editor interface. At the top, there's a toolbar with various icons for file operations like Open, Save, Undo, Redo, and Print. Below the toolbar, a row of five preview cards is displayed, each showing a different design template: 'Slate' (dark theme), 'MERLOT' (green border), 'Time Machine' (blue gradient), 'MINIMAL' (light gray with horizontal lines), and 'Leap Day' (yellow background). To the right of these cards are arrows for navigating through the designs. Further right are buttons for 'HIDE', 'EDIT', and 'PUBLISH' with a checkmark icon.

The main content area features a large red banner with white text that reads 'O2PLATFORM.GITHUB.COM' and 'Automating Application Security Knowledge and Workflows'. Below the banner, a blue header bar contains the text 'Welcome to the OWASP O2 Platform Website'. The main body text describes the O2 platform as a paradigm for performing, documenting, and distributing Web Application security reviews, designed to automate knowledge and workflows and allow non-security experts to access and consume security knowledge. It encourages joining the O2 Platform Mailing List for questions.

Welcome to the OWASP O2 Platform Website

The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to **Automate Application Security Knowledge and Workflows** and to **Allow non-security experts to access and consume Security Knowledge**.

Please join the [O2 Platform Mailing List](#) and ask questions! **You will be lost when trying to use O2!** So instead of giving up, ask for help :)

Time Machine:

The screenshot shows the Wix website builder interface. At the top, there's a navigation bar with icons for undo, redo, save, and publish. Below the navigation is a row of theme preview cards: 'Slate' (dark grey), 'MERLOT' (light beige with a ribbon banner), 'Time Machine' (green border with a blue ribbon banner), 'Minimal' (grey with horizontal lines), and 'Leap Day' (blue with a yellow banner). The 'Minimal' theme is highlighted with a green border. On the right side of the interface, there are buttons for 'HIDE', 'EDIT', and 'PUBLISH'. The main content area displays the website for 'O2platform.github.com' with the title 'Automating Application Security Knowledge and Workflows' and a button to 'View O2platform.github.com on GitHub'. The page content includes a heading 'Welcome to the OWASP O2 Platform Website', a paragraph about the O2 platform's purpose, and a message encouraging users to join the mailing list.

O2platform.github.com

Automating Application Security Knowledge and Workflows

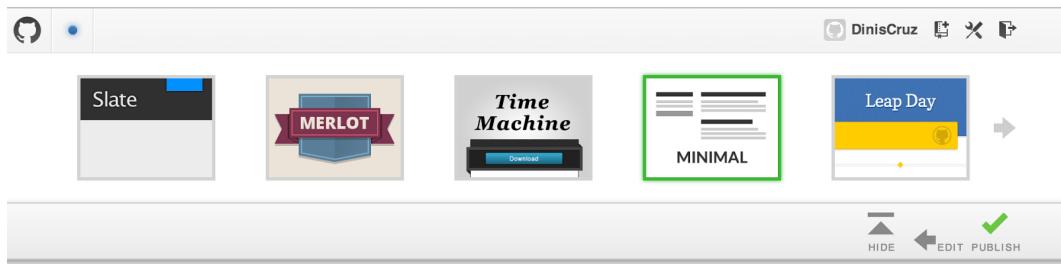
</> View O2platform.github.com on GitHub

Welcome to the OWASP O2 Platform Website

The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to Automate Application Security Knowledge and Workflows and to Allow non-security experts to access and consume Security Knowledge.

Please join the [O2 Platform Mailing List](#) and ask questions! You will be lost when trying to use O2!
So instead of giving up, ask for help :)

Minimal:



O2platform.github.com

Automating Application Security Knowledge
and Workflows

[View My GitHub Profile](#)

Welcome to the OWASP O2 Platform Website

The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to **Automate Application Security Knowledge and Workflows** and to **Allow non-security experts to access and consume Security Knowledge**.

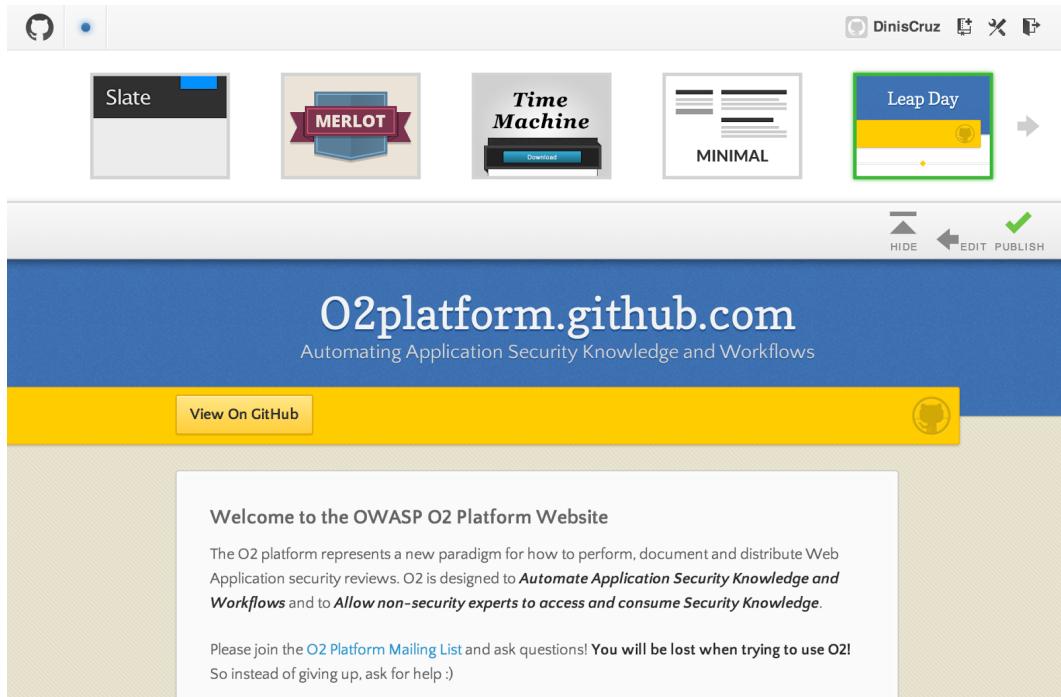
Please join the [O2 Platform Mailing List](#) and ask questions! **You will be lost when trying to use O2!** So instead of giving up, ask for help :)

Download Stand-Alone Application

You can run all O2 Tools and Scripts using the stand-alone executable which you can download from:

- [O2 Platform - Main O2 Gui v5.1.exe](#)
- [O2 Platform - Main O2 Gui v5.1.1 - x64.exe](#)

Leap Day:



Modernist:

The screenshot shows a GitHub repository interface. At the top, there are five theme preview cards: 'MODERNIST' (blue), 'hack' (dark blue), 'Midnight' (black), 'Architect' (blue), and 'Tactile' (light gray). Below these are two buttons: 'HIDE' and 'EDIT PUBLISH'. The main content area displays the repository's homepage for 'O2platform.github.com'. The header includes the repository name, a brief description ('Automating Application Security Knowledge and Workflows'), and a 'View On GitHub' button. The main content features a heading 'Welcome to the OWASP O2 Platform Website' and a paragraph explaining the platform's purpose: 'The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to **Automate Application Security Knowledge and Workflows** and to **Allow non-security experts to access and consume Security Knowledge**.'. It also encourages users to join the mailing list.

** Hack:**

The screenshot shows the GitHub themes interface. At the top, there are five theme cards: 'MODERNIST', 'hack', 'Midnight', 'Architect', and 'Tactile'. The 'Midnight' card is highlighted with a green border. Below the cards, the URL `./ 02platform.github.com` is displayed in green. The page title is "Automating Application Security Knowledge and Workflows". A "View on GitHub" button is present. On the right side, there are buttons for "HIDE", "EDIT", and "PUBLISH". The main content area displays the text: "Welcome to the OWASP 02 Platform Website". It continues: "The 02 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. 02 is designed to **Automate Application Security Knowledge and Workflows** and to **Allow non-security experts to access and consume Security Knowledge**.
The text is in white on a dark background.

Midnight:

The screenshot shows the GitHub theme selection interface. At the top, there are five theme preview cards: MODERNIST, hack, Midnight, Architect (which is highlighted with a green border), and Tactile. Below the cards is a navigation bar with icons for HIDE, EDIT, and PUBLISH. A large button labeled "View On GitHub" is prominently displayed. The main content area features the "Architect" theme's design, which has a dark background with light-colored text and a clean layout. The URL "O2platform.github.com" and the tagline "Automating Application Security Knowledge and Workflows" are visible at the top of the page. Below this, there is a welcome message and a note about the O2 platform's purpose.

MODERNIST
A Theme for Code
NSLog(@"Hello, World");
Designed with a modern touch.

hack

Midnight
A bit of the GitHub spice.

Architect

Tactile
It just feels right.

HIDE EDIT PUBLISH

View On GitHub

O2platform.github.com

Automating Application Security Knowledge and Workflows

Project maintained by [o2platform](#)

Hosted on GitHub Pages — Theme by [mattgraham](#)

Welcome to the OWASP O2 Platform Website

The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to **Automate Application Security Knowledge and Workflows** and to **Allow non-security experts to access and consume Security Knowledge**.

Please join the [O2 Platform Mailing List](#) and ask questions! **You will be lost when trying to use O2!** So instead of giving up, ask for help :)

Architect:

The screenshot shows the GitHub Pages interface for the repository [O2platform/O2Platform](https://github.com/O2platform/O2Platform). At the top, there's a navigation bar with icons for search, refresh, and user profile (DinisCruz). Below the navigation is a horizontal menu bar with links for MODERNIST, hack, Midnight, Architect (which is highlighted in blue), and Tactile. To the right of the menu are buttons for HIDE, EDIT, and PUBLISH, along with a green checkmark icon.

The main content area features a large blue header with the text "O2platform.github.com" in white. Below the header, it says "Automating Application Security Knowledge and Workflows". On the right side of the header is a GitHub button with the text "Follow me on GitHub" and the GitHub logo.

In the bottom right corner of the main content area, there's a small note: "This page was generated by GitHub Pages using the Architect theme by Jason Long."

The main content area contains two sections:

- Welcome to the OWASP O2 Platform Website**: A section with a light gray background containing text about the O2 platform and a call to join the mailing list.
- Please join the O2 Platform Mailing List**: A section with a light gray background containing text encouraging users to join the mailing list.

Tactile:

The screenshot shows a GitHub repository page for "O2platform.github.com". At the top, there's a navigation bar with icons for repository management (star, fork, etc.) and a user profile for "DinisCruz". Below the header, there are five preview cards for different themes: "MODERNIST", "hack", "Midnight", "Architect", and "Tactile". The "Tactile" card is highlighted with a green border. On the right side of the page, there are buttons for "HIDE", "EDIT", and "PUBLISH". The main content area features the repository name "O2platform.github.com" in large bold letters, followed by the subtitle "Automating Application Security Knowledge and Workflows". Below this, there's a "View on GitHub" button with a cartoon character icon. A section titled "Welcome to the OWASP O2 Platform Website" contains text about the platform's purpose: "The O2 platform represents a new paradigm for how to perform, document and distribute Web Application security reviews. O2 is designed to **Automate Application Security Knowledge and Workflows** and to **Allow non-security experts to access and consume Security Knowledge**". At the bottom, there's a note encouraging users to join the mailing list: "Please join the [O2 Platform Mailing List](#) and ask questions! You will be lost when trying to use O2! So instead of giving up, ask for help :)".

And finally Dinky

The screenshot shows the Wix website builder interface with the following details:

- Header:** Shows the URL "o2platform.github.io" and navigation icons.
- Left Sidebar:** Contains the title "O2 Platform" and a sub-section "Automating Application Security Knowledge and Workflows". Below it is a red button labeled "GitHub Profile".
- Content Area:**
 - Welcome to the OWASP O2 Platform Website:** A heading with a descriptive paragraph about the O2 platform's paradigm and goals.
 - Download Stand-Alone Application:** A section with a link to download the executable.
 - Links:** A list of two download links: "O2 Platform - Main O2 Gui v5.1.exe" and "O2 Platform - Main O2 Gui v5.1.1 - x64.exe".
- Footer:** Standard Wix interface elements including "HIDE", "EDIT", and "PUBLISH" buttons.

which was my preferred choice , so I clicked on **Publish**, and here it is in action (using the default domain provided by GitHub)** **

The screenshot shows the published website at o2platform.github.io:

- Header:** Shows the URL "o2platform.github.io" and navigation icons.
- Content Area:**
 - Welcome to the OWASP O2 Platform Website:** A heading with a descriptive paragraph about the O2 platform's paradigm and goals.
 - Download Stand-Alone Application:** A section with a link to download the executable.
 - Links:** A list of two download links: "O2 Platform - Main O2 Gui v5.1.exe" and "O2 Platform - Main O2 Gui v5.1.1 - x64.exe".

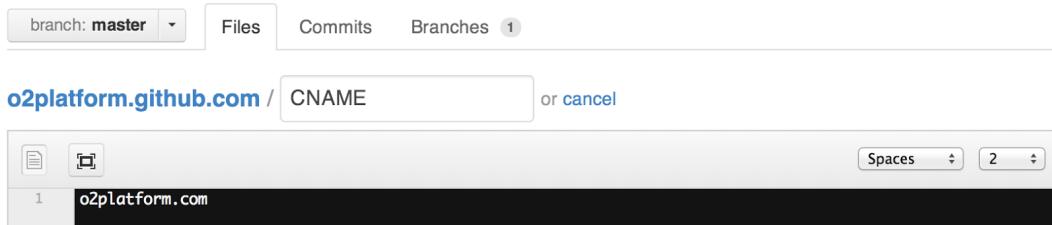
Next step was to set up custom domain, which is explained here:

The screenshot shows a browser window displaying the GitHub Pages help article titled "Setting up a custom domain with Pages". The URL in the address bar is <https://help.github.com/articles/setting-up-a-custom-domain-with-pages>. The page content includes a search bar and a "How can we help?" button. The main heading is "Setting up a custom domain with Pages". Below it, a sub-section title is "Setting the domain in your repo". A tip box contains the following text: "Let's say you own the domain [example.com], and you would like to use it for your Pages. Telling the GitHub server to serve from this domain is easy: just create a file named [CNAME] in the root of your pages and put the domain (or subdomain) into the file: example.com". A tip note states: "Tip: After creating the file, it can take up to **10 minutes** for the content to become available. Once the Page content is available, you can continue to make the DNS changes in the next step. If the Page doesn't build successfully, you'll receive a notification from GitHub."

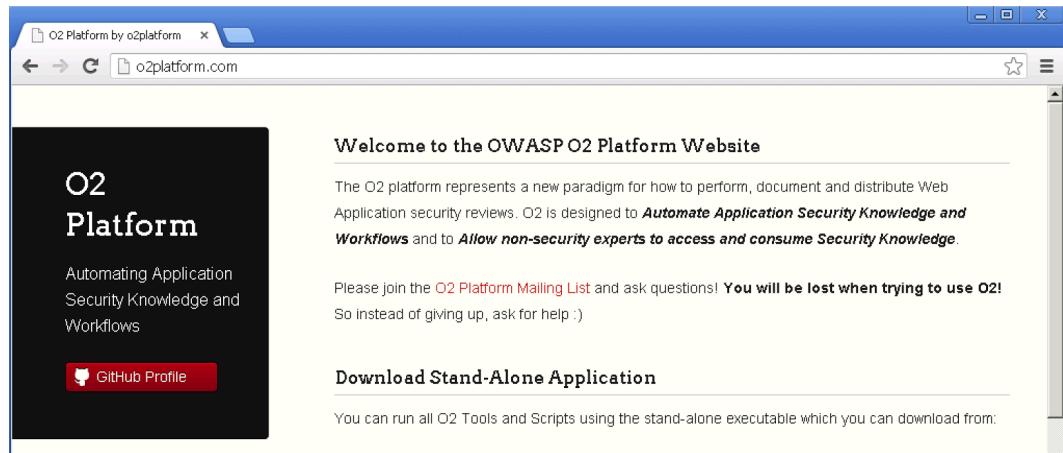
and is basically a case of creating new file:

The screenshot shows a GitHub repository page for "o2platform.github.com". The top navigation bar includes options for "Clone in Mac", "ZIP", "HTTP", "SSH", "Git Read-Only", and a copy link. The repository details show "branch: master" and a "Create a new file here" button. The main content area displays a commit by "DinisCruz" from 6 minutes ago, which adds an "images" folder. The commit message is "Replace master branch with page content via GitHub".

Called CNAME, with the contents of the domain



After saving it (and local cache flushed), the GitHub pages now point to <http://o2platform.com>⁴¹



⁴¹<http://o2platform.com/>

8. June 2013

- Creating TeamMentor release 3.3.2 (3.2 version with HotFix 2)
- Fixing a couple bugs and pushing new TeamMentor 3.4 Dev Version (from 4 to 5)
- Gource Visualisation of “TeamMentor Git Development - 18 Months in 180 Seconds”

Creating TeamMentor release 3.3.2 (3.2 version with HotFix 2)

Now that the two P0 issues are marked as fixed (after a round of QA):

The screenshot shows a list of two issues in a bug tracking system. Both issues are marked as 'P0' priority, 'Resolution: Fixed', 'Area: Backend', and 'Type: Bug'. The first issue was opened by roman87 25 days ago and has 8 comments. The second issue was opened by DinisCruz a month ago and also has 8 comments. The interface includes filters for 'Open' and 'Closed' issues, sorting by 'Newest', and buttons for 'Close', 'Label', 'Assignee', and 'Milestone'.

Issue #	Description	Priority	Resolution	Area	Type	Comments
#507	Unable to delete a folder	P0	Resolution: Fixed	Backend	Bug	8
#482	Article Save problem on Libraries whose name didn't match (when using git sync)	P0	Resolution: Fixed	Backend	Bug	8

It's time to publish the 3.3.2 TeamMentor release.

At the moment the code changes are in the 3_3_2_HotFix branch

The screenshot shows the GitHub repository page for 'TeamMentor / Master'. The 'Code' tab is selected. The repository has 296 issues and 0 pull requests. The 'branch: 3_3_2_HotFix' dropdown is open. The page includes links for 'Clone in Windows', 'ZIP', 'HTTP', 'SSH', and 'Git Read-Only', along with the URL 'git@github.com:TeamMentor/Master.git'. It also shows 'Read-Write access' and a 'Tags' section.

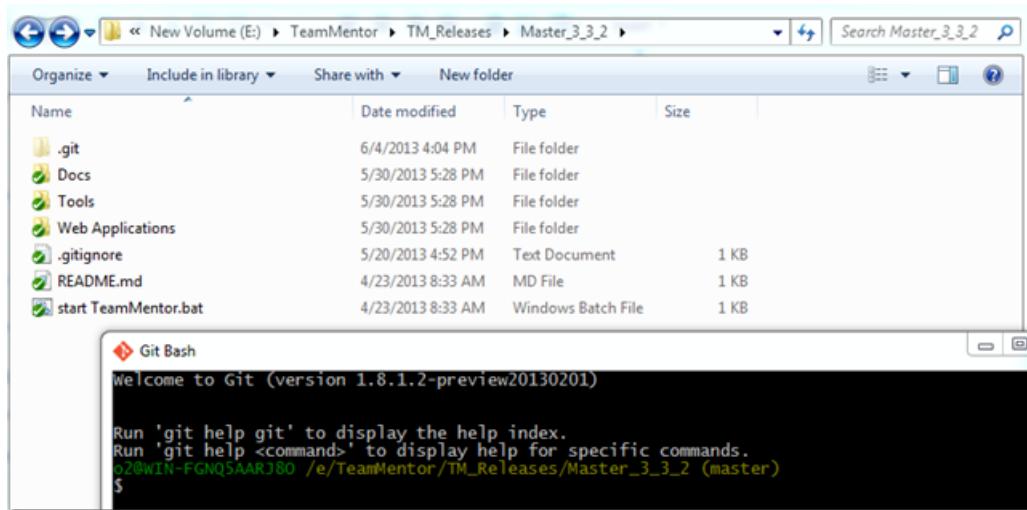
Which contains the commits that made up the 3.3.2 – RC2 version (with the last commit being the [852d877290](#))

The screenshot shows the GitHub Commit History for the 'Master' branch. It displays four commits:

- Jun 04, 2013:** Fix issue where Library renames were not being persisted to disk. (Commit 852d877290) - authored by DinisCruz 6 days ago.
- May 30, 2013:**
 - Setting version to 3.3.2 - RC1 (Commit 233435cc) - authored by DinisCruz 11 days ago.
 - Merge remote-tracking branch 'dinis/482_LibraryRename' (Commit b694fe3780) - authored by DinisCruz 11 days ago.
 - Merge remote-tracking branch 'dinis/507_FolderDelete' (Commit 278612d52c) - authored by DinisCruz 11 days ago.

As set in our release process, to make this the official official version, I will remove the RC2 bit from the version number and make it the final commit for this release.

So I opened a Git Bash on a local copy of that repo TeamMentor/Master (same thing as doing a git clone and pull of the 3_3_2_HotFix branch)

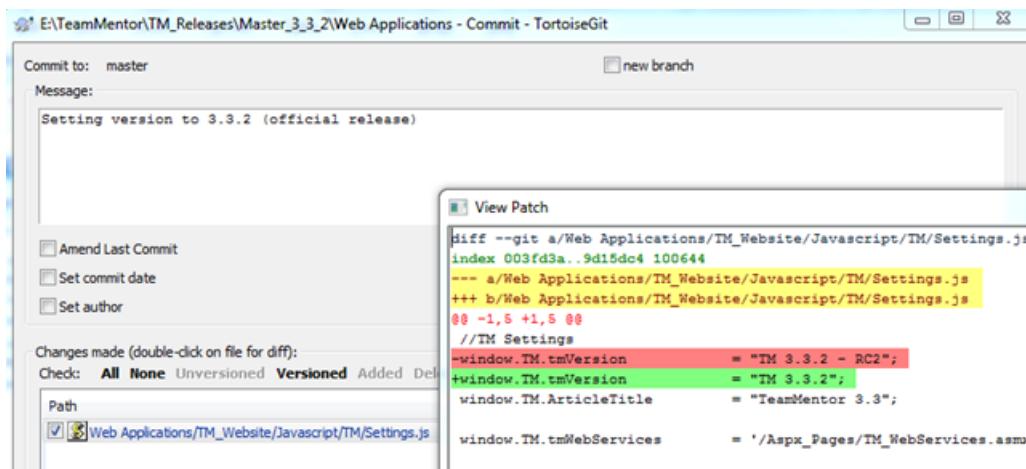


A quick look using `git log --decorate --graph --oneline --date-order` shows that the latest commit is `852d877` (which matches the version at GitHub that was QAed and checked for this release)

```
$ git log --decorate --graph --oneline --date-order
* 852d87 (HEAD, azure/master, master) Fix issue where Library renames where not being persisted
  3343 (dev/master) Setting version to 3.3.2 - RC1
* b694fc3 Merge remote-tracking branch 'dinis/482_LibraryRename'
  278612d Merge remote-tracking branch 'dinis/507_FolderDelete'
* c599895 Merge remote-tracking branch 'dinis/524_SetUserRole'
* | 4a34057 (dinis/524_SetUserRole) Fixing bug that prevented the user details save from the
* | fe2b72f (dinis/482_LibraryRename) Fixed the issue https://github.com/TeamMentor/Master/issu
  00c304a Updating ga.js
* f7335d0 (dinis/507_FolderDelete) Fixed xmlDB_Delete_Folder method to correctly handle folder
* a9eb167 (tag: v3.3.1, dinis/master, dinis/HotFix_3_3_1, dev/HotFix_3_3_1, 437-Password-Expiry)
* d0085dc TM 3.3.1 - Dev.3: removed 'raw' view from User_Edit.cshtml
```

My next steps was to change the version number,

... commit that small change:



... add the v3.3.2 tag (see [Adding Tags to TeamMentor Master repository¹](#) for more details on tagging)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Releases/Master_3_3_2 (master)
$ git tag -a v3.3.2 -m '3.3 HotFix 2'
```

... and pushed into TeamMentor/Master the commit and tag:

¹<https://www.blogger.com/blog.diniscruz.com/2012/10/adding-tags-to-teammentor-master.html>

```
$ git push git@github.com:TeamMentor/Master.git master:master
Counting objects: 13, done.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 565 bytes, done.
Total 7 (delta 6), reused 0 (delta 0)
To git@github.com:TeamMentor/Master.git
  a9eb167..6f63a57  master -> master
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Releases/Master_3_3_2 (master)
$ git push git@github.com:TeamMentor/Master.git v3.3.2
Counting objects: 1, done.
Writing objects: 100% (1/1), 166 bytes, done.
Total 1 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Master.git
 * [new tag]           v3.3.2 -> v3.3.2
```

Just to confirm, let's take a look at GitHub:

Main page shows the 3.3.2 commit

The screenshot shows the GitHub repository page for 'Master'. At the top, there are navigation tabs: 'branch: master' (selected), 'Files', 'Commits', 'Branches', and 'Tags'. The 'Commits' tab is active, showing a list of recent commits. The first commit in the list is highlighted with a blue border. The commit details are as follows:

Setting version to 3.3.2 (official release)		
DinisCruz	authored 4 minutes ago	latest commit 6f63a5723a
Docs	8 months ago	adding v3.1 of the ReleaseNotes v3.1.txt [DinisCruz]
Tools	9 months ago	adding pinned functionality to Applied filters [DinisCruz]
Web Applications	4 minutes ago	Setting version to 3.3.2 (official release) [DinisCruz]
.gitignore	2 months ago	DeTabify file and added _Customizations folder to .gitignore (that co... [DinisCruz]
README.md	8 months ago	added ReleaseNotes for v3.2 and moved to /Docs folder [DinisCruz]
start TeamMentor bat	8 months ago	Set version to 3.2 [DinisCruz]

... so does the commit page:

Master / Commit History [RSS](#)

Jun 10, 2013

Setting version to 3.3.2 (official release)
DinisCruz authored 4 minutes ago

Jun 04, 2013

Fix issue where Library renames where not being persisted to disk:
DinisCruz authored 6 days ago

May 30, 2013

Setting version to 3.3.2 - RC1
DinisCruz authored 11 days ago

Keyboard shortcuts available [? \[Esc\]](#)

6f63a5723a +
Browse code ↗

852d877290 +
Browse code ↗

b4233435cc +
Browse code ↗

... and the Tags view:

Tags [? \[Esc\]](#)

Tags

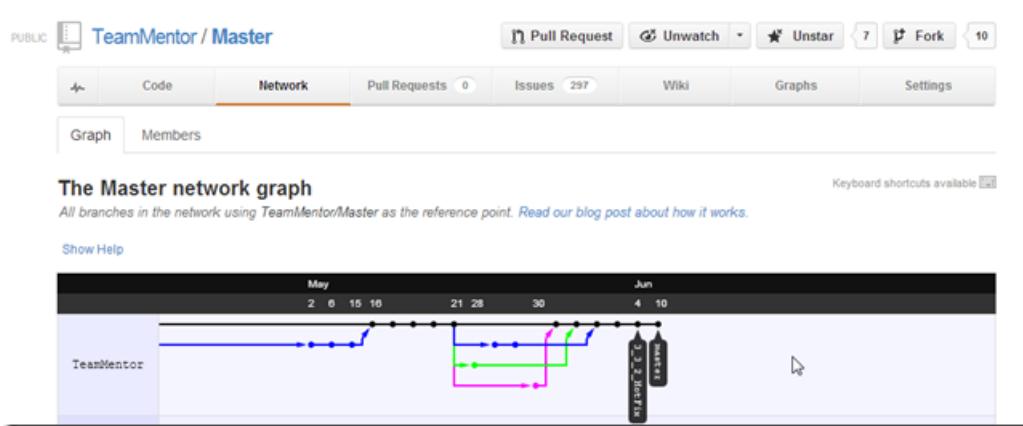
4 minutes ago **v3.3.2 - 3.3 HotFix 2**
6f63a57 zip tar.gz

20 days ago **v3.3.1 - 3.3 HotFix 1**
a9eb167 zip tar.gz

2 months ago **v3.3 - 3.3 Release**
e004e1d zip tar.gz

6 months ago **v3.2.4 - 3.2.4 Release**
4e39135 zip tar.gz

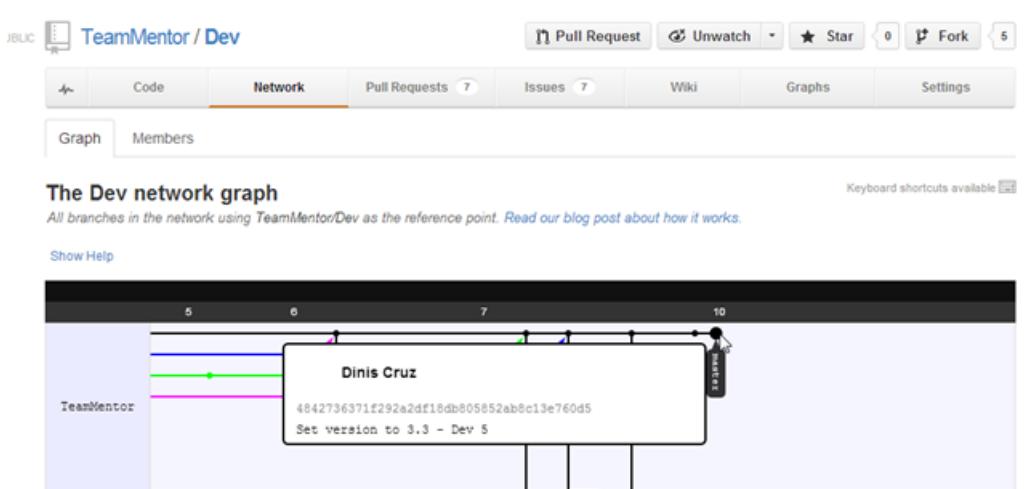
...and the Network graph:



The deployment of this version is now passed to the hands of the Infrastructure team, which will update all TM sites currently managed by SI.

On the development side, the last thing to do, is to add this version to the TeamMentor/Dev fork so that it is part of the next release

At the moment TeamMentor/Dev is on this commit



In a local copy of the this repo, I did a pull from TeamMentor/Master

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git pull git@github.com:TeamMentor/Master.git master:master
remote: Counting objects: 35, done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 22 (delta 19), reused 8 (delta 5)
Unpacking objects: 100% (22/22), done.
From github.com:TeamMentor/Master
  ! [rejected]           master      -> master  (non-fast-forward)
 * [new tag]            v3.3.2     -> v3.3.2
```

... which failed (on master) because there has been updates done on this repo (since the last merge).

So I created a new branch called *3_3_2_merge*

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git checkout -b 3_3_2_merge
Switched to a new branch '3_3_2_merge'
```

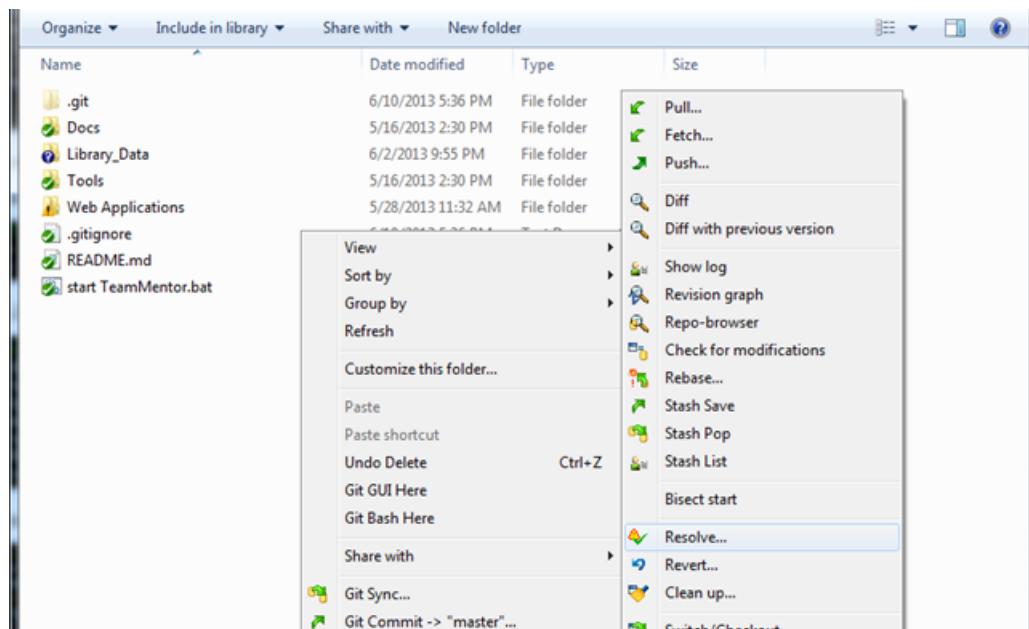
Forced pulled the 3.3.2 code into it (the code from TeamMentor/Master)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3_3_2_merge)
$ git pull -f git@github.com:TeamMentor/Master.git master:3_3_2_merge
From github.com:TeamMentor/Master
 + 4842736...6f63a57 master      -> 3_3_2_merge (forced update)
Warning: fetch updated the current branch head.
Warning: fast-forwarding your working tree from
Warning: commit 4842736371f292a2df18db805852ab8c13e760d5.
Already up-to-date.
```

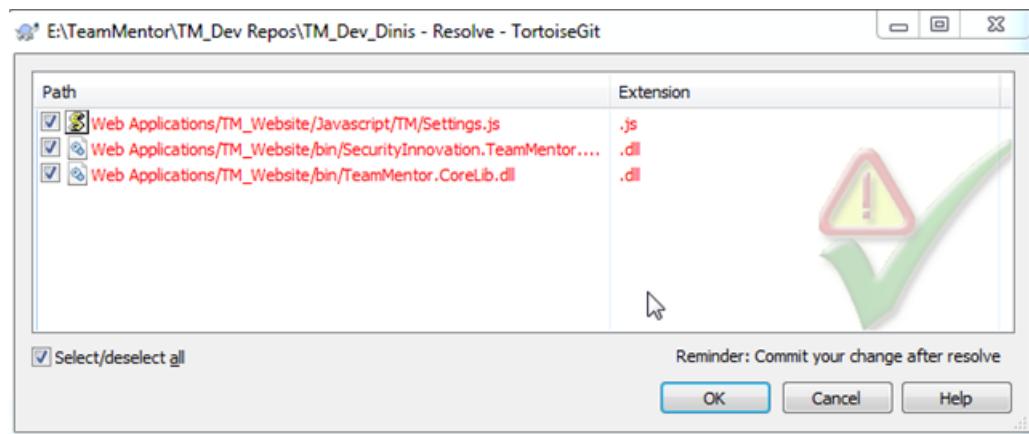
switch back into master branch and merged with *3_3_2_merge* branch

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3_3_2_merge)
$ git checkout master
Switched to branch 'master'
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git merge 3_3_2_merge
warning: Cannot merge binary files: Web Applications/TM_Website/bin/TeamMentor.CoreLib.dll (HEAD
warning: Cannot merge binary files: Web Applications/TM_Website/bin/SecurityInnovation.TeamMentor
Auto-merging Web Applications/TM_Website/bin/TeamMentor.CoreLib.dll
CONFLICT (content): Merge conflict in Web Applications/TM_Website/bin/TeamMentor.CoreLib.dll
Auto-merging Web Applications/TM_Website/bin/SecurityInnovation.TeamMentor.Website.dll
CONFLICT (content): Merge conflict in Web Applications/TM_Website/bin/SecurityInnovation.TeamMent
Auto-merging Web Applications/TM_Website/Javascript/TM/Settings.js
CONFLICT (content): Merge conflict in Web Applications/TM_Website/Javascript/TM/Settings.js
Automatic merge failed; fix conflicts and then commit the result.
```

...which had (as expected) a couple conflicts), with I'm going to resolve using



... which lists the conflicts



... and in this case was mainly the version number:

Their - REMOTE

```

1 //TM-Settings
2 window.TM.tmVersion.....="TM-3.3.2--RC1";  

3 window.TM.ArticleTitle.....="TeamMentor-3.3";  

4  

5 window.TM.WebServices.....='/Aspx_Pages/TM_Nebs';  

6 window.TM.NotAuthorizedPage.....'/Html_Pages/Gui/Pan  

7  

8 window.TM.Tracking_Google_Analytics_ID.....  

9 window.TM.Gui.showLibraryStructureToAnonymous.....  

10 window.TM.Gui.LoadLibraryData.....
```

Mine - LOCAL

```

1 //TM-Settings
2 window.TM.tmVersion.....="TM-3.3.2--RC1";  

3 window.TM.ArticleTitle.....="TeamMentor-3.3";  

4  

5 window.TM.WebServices.....='/Aspx_Pages/TM_Nebs';  

6 window.TM.NotAuthorizedPage.....'/Html_Pages/Gui/Pan  

7  

8 window.TM.Tracking_Google_Analytics_ID.....  

9 window.TM.Gui.showLibraryStructureToAnonymous.....  

10 window.TM.Gui.LoadLibraryData.....
```

Merged - Settings.js

```

1 //TM-Settings
2 window.TM.tmVersion.....="TM-3.3.2--RC1";  

3 window.TM.ArticleTitle.....="TeamMentor-3.3";  

4  

5 window.TM.WebServices.....='/Aspx_Pages/TM_Nebs';  

6 window.TM.NotAuthorizedPage.....'/Html_Pages/Gui/Pan  

7  

8 window.TM.Tracking_Google_Analytics_ID.....  

9 window.TM.Gui.showLibraryStructureToAnonymous.....  

10 window.TM.Gui.LoadLibraryData.....
```

... easily fixed by making the local version the one to use:

Their - REMOTE

```

1 //TM-Settings
2 window.TM.tmVersion.....="TM-3.3.2--RC1";  

3 window.TM.ArticleTitle.....="TeamMentor-3.3";  

4  

5 window.TM.WebServices.....='/Aspx_Pages/TM_Nebs';  

6 window.TM.NotAuthorizedPage.....'/Html_Pages/Gui/Pan  

7  

8 window.TM.Tracking_Google_Analytics_ID.....  

9 window.TM.Gui.showLibraryStructureToAnonymous.....  

10 window.TM.Gui.LoadLibraryData.....
```

Mine - LOCAL

```

1 //TM-Settings
2 window.TM.tmVersion.....="TM-3.3.2--RC1";  

3 window.TM.ArticleTitle.....="TeamMentor-3.3";  

4  

5 window.TM.WebServices.....='/Aspx_Pages/TM_Nebs';  

6 window.TM.NotAuthorizedPage.....'/Html_Pages/Gui/Pan  

7  

8 window.TM.Tracking_Google_Analytics_ID.....  

9 window.TM.Gui.showLibraryStructureToAnonymous.....  

10 window.TM.Gui.LoadLibraryData.....
```

Merged - Settings.js

```

1 //TM-Settings
2 window.TM.tmVersion.....="TM-3.3.2--RC1";  

3 window.TM.ArticleTitle.....="TeamMentor-3.3";  

4  

5 window.TM.WebServices.....='/Aspx_Pages/TM_Nebs';  

6 window.TM.NotAuthorizedPage.....'/Html_Pages/Gui/Pan  

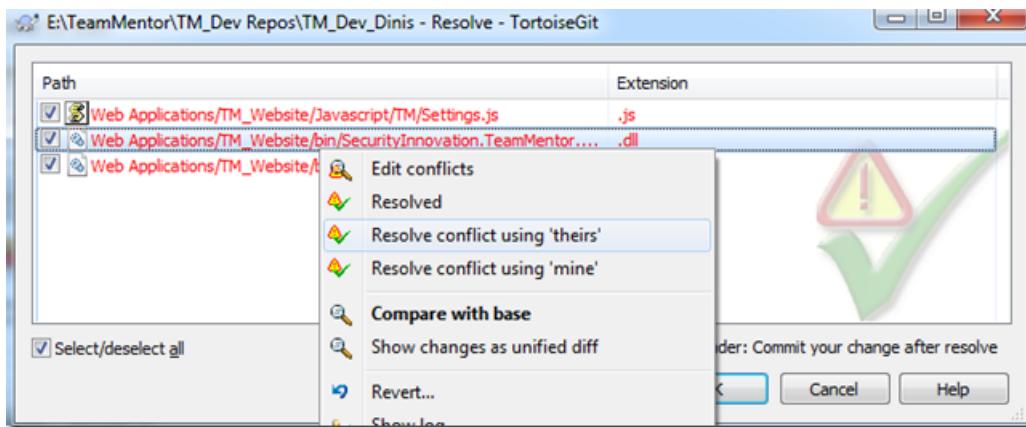
7  

8 window.TM.Tracking_Google_Analytics_ID.....  

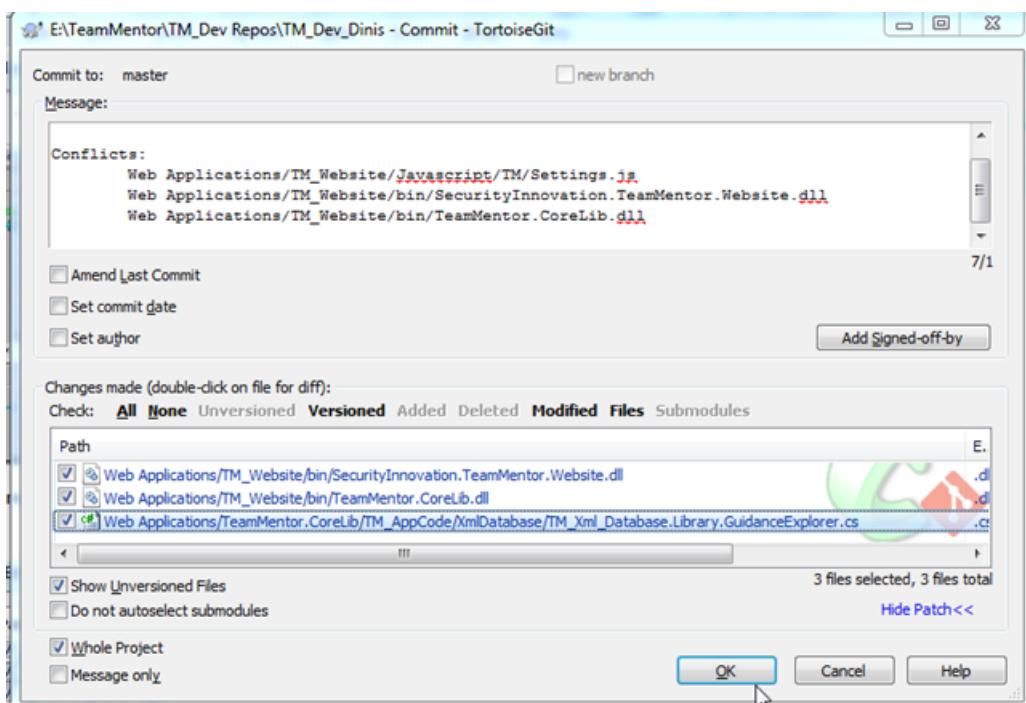
9 window.TM.Gui.showLibraryStructureToAnonymous.....  

10 window.TM.Gui.LoadLibraryData.....
```

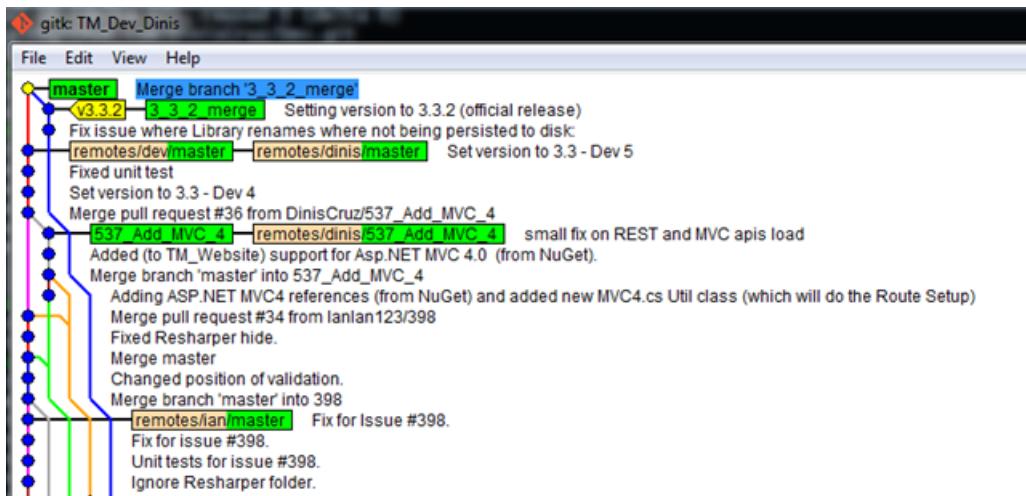
... next I resolved the dlls by selecting one of them (doesn't really matter since they will be recompiled soon)



... and committed the merge:



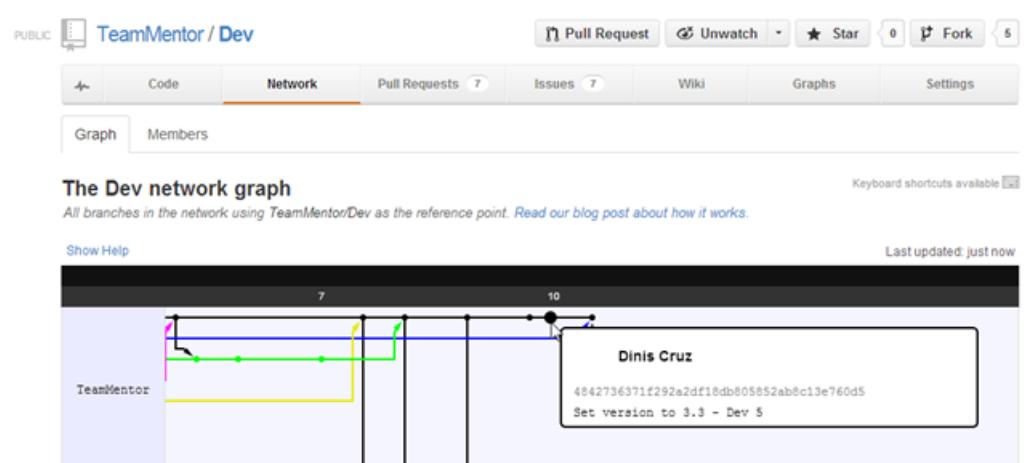
....which we can now see on Gitk:



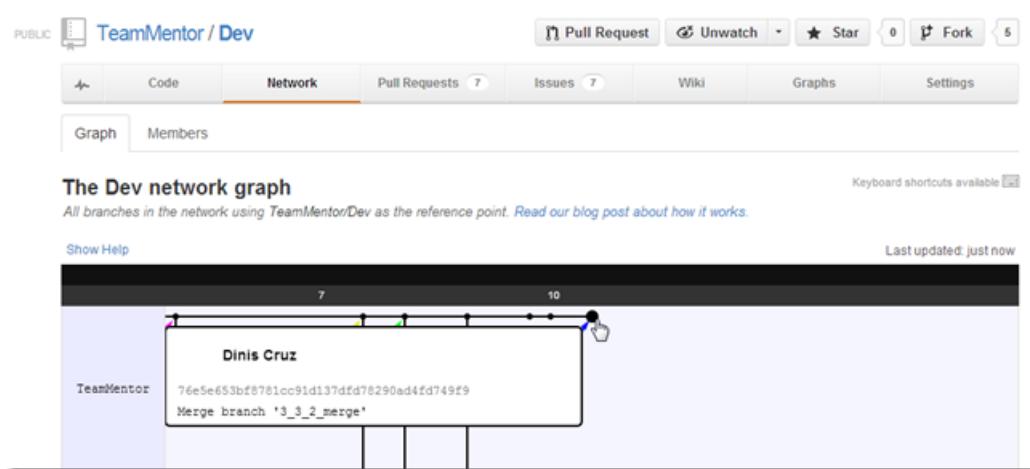
The final step is to push these commits into TeamMentor/Dev

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git push dev master:master
Counting objects: 54, done.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (29/29), 4.10 KiB, done.
Total 29 (delta 26), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
    4842736371f292a2df18db805852ab8c13e760d5
        master -> master
```

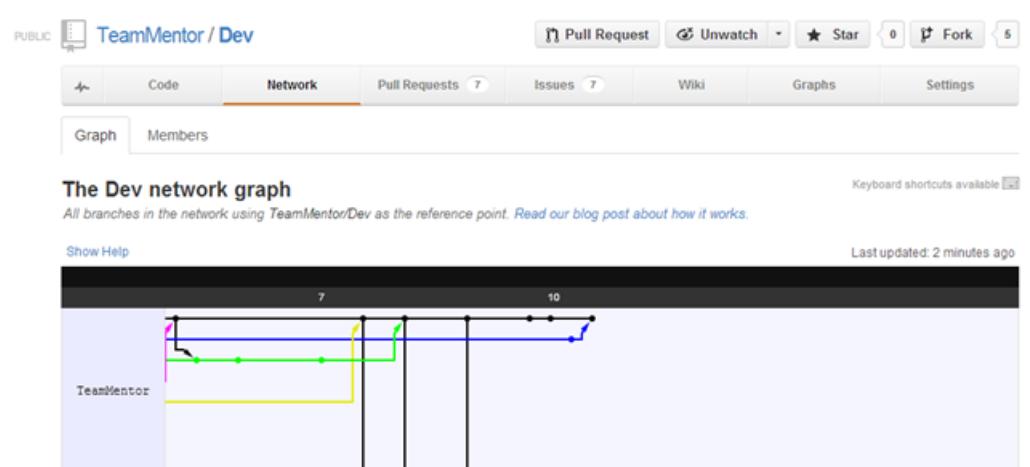
.... which can be seen on the following couple graphs:



The image above shows the TeamMentor/Dev commit done before the 3.3.2 merge, and below is the last commit made



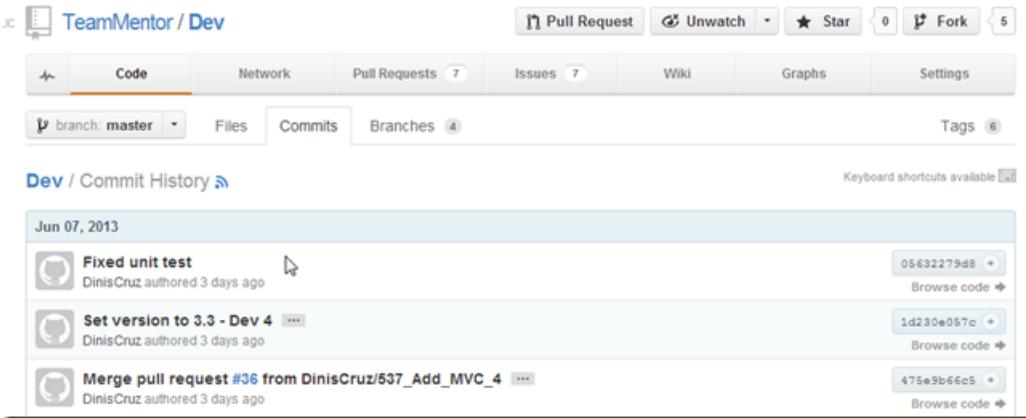
It might be easier to read with out the labels (in blue is the *TeamMentor/Dev* code in black is the *_TeamMentor/Master_* code:



Fixing a couple bugs and pushing new TeamMentor 3.4 Dev Version (from 4 to 5)

This post shows one way to use GitHub to update the main development branch of TeamMentor.

At the moment [TeamMentor/Dev](https://github.com/TeamMentor/Dev)² repo is at version *3.3 – Dev 4*



Jun 07, 2013

- Fixed unit test (DinisCruz, 3 days ago) → 05632279db → Browse code
- Set version to 3.3 - Dev 4 (DinisCruz, 3 days ago) → 1d230e057c → Browse code
- Merge pull request #36 from DinisCruz/537_Add_MVC_4 (DinisCruz, 3 days ago) → 475e9b66c5 → Browse code

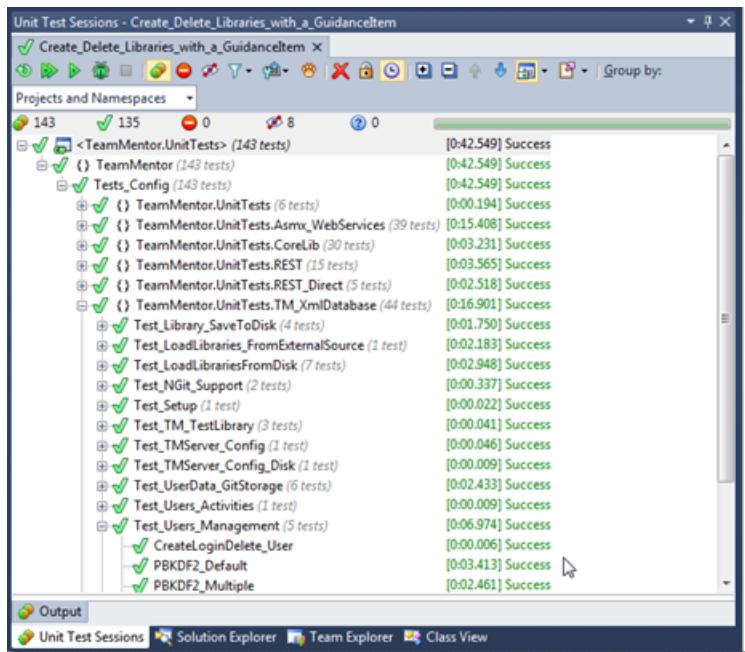
This version was pushed on Friday, and it introduced a nasty side effect on the wsdl generation (see issue [546³](#)) and a minor bug in the user edit (from the old control panel).

Both probs were picked up by Michael's TM UI Unit Tests, which is another good example of the power of that type of UI/Browser integration tests.

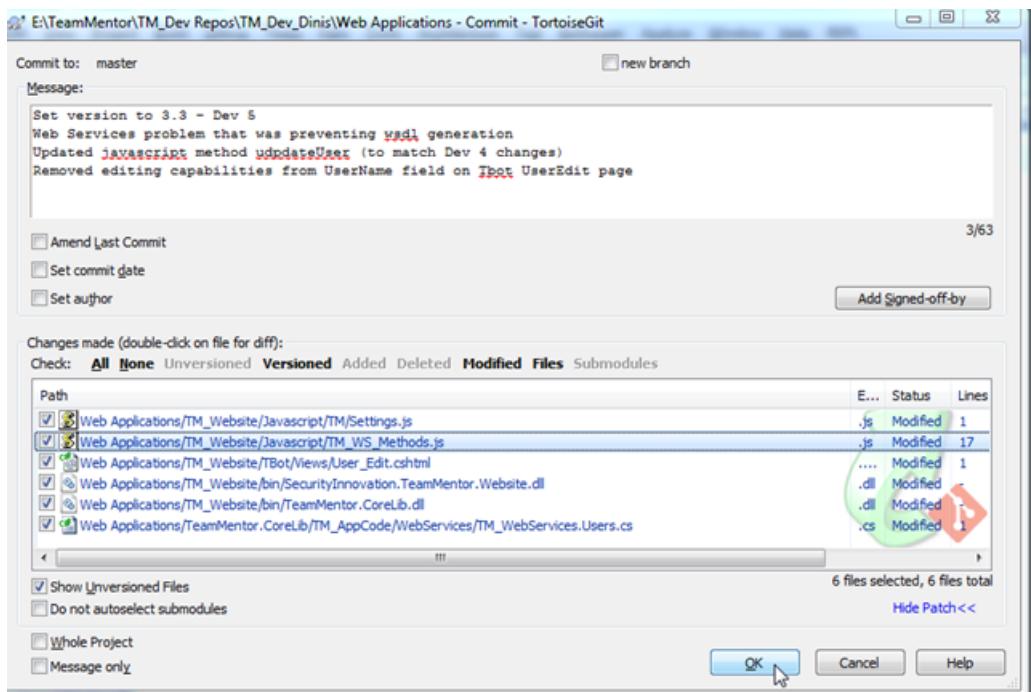
So, I went into my local dev repo, made the code fixes, checked that they were working and executed all unit tests (to make sure we are still good)

²<https://github.com/TeamMentor/Dev>

³<https://github.com/TeamMentor/Master/issues/546>



Next I committed the changes locally:



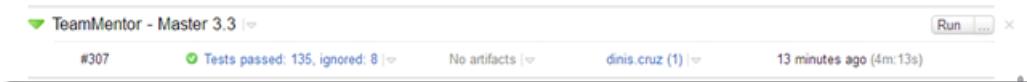
And pushed the commit to the main Dev repo and my personal Dev repo (this is the same as doing a Pull Request and authorizing it via the GUI):

```
02@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git push dev master:master
Counting objects: 35, done.
Compressing objects: 100% (18/18), done.
Writing objects: 100% (18/18), 74.35 KiB, done.
Total 18 (delta 17), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
  0563227..4842736  master -> master
02@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git push dinis master:master
Counting objects: 35, done.
Compressing objects: 100% (18/18), done.
Writing objects: 100% (18/18), 74.35 KiB, done.
Total 18 (delta 17), reused 0 (delta 0)
To git@github.com:DinisCruz/Dev.git
  0563227..4842736  master -> master
```

Since TeamCity is configured to run on commit the master branch of TeamMentor/Dev a quick look at its web interface shows that the build started



... after 4m:13s the build was completed, with all unit tests are passed and



... and the Dev QA site pushed into Azure:

The screenshot shows the TeamMentor web interface. On the left, there's a sidebar with 'Applied Filters' and 'Guidance Libraries' containing items like 'New_Library_0494', 'OWASP', 'PDF Policy', 'PDF Policy Sections', 'Test Library', and 'Top Vulnerabilities'. The main content area has a search bar and two filter panels: 'Technology' (with checkboxes for Any, ASP.NET 3.5, Java) and 'Phase' (with checkboxes for Any, Deployment, Design, Implementation). Below these is a table titled 'Showing 174 items (out of 174)' with columns for Title, Technology, and Phase. The first row in the table is 'A Secure Key Storage Location Is Used' (Technology: ASP.NET 3.5, Phase: Implementation). To the right, a 'Selected Article' panel displays the title 'A Secure Key Storage Location Is Used' and a section 'What to Check For' with the text: 'Ensure that application keys are stored in a well defined location, such as the encrypted sections of the application's web.config or encrypted in the Windows Registry. Verify that they are not hard coded into the application or stored as plaintext inside configuration files.' There's also a 'Why' section.

Gource Visualization of “TeamMentor Git Development - 18 Months in 180 Seconds”

Here is a pretty cool video of 18 months of Git commits using the [Gource tool](#)⁴ which is a software version control visualization tool.

Try to see it in full screen and at 1024p HD quality:

Here are the [Gource settings](#)⁵ used to create this video:

```
gource.exe -s 1 -file-idle-time 0 -key -title "TeamMentor" -font-size 30 - hide  
dirnames -date-format "%d/%b/%y" -bloom-multiplier 0.5 -bloom-intensity 0.5  
-f
```

Since the original video was 10 minutes, I used Camtasia *Clip Speed* feature to compress it to about 3 minutes (~180 sec)

⁴<https://code.google.com/p/gource/>

⁵<https://code.google.com/p/gource/wiki/Controls>

9. August 2013

- Creating a clone of WebGoat on GitHub

Creating a clone of WebGoat on GitHub

I needed a couple vulnerable source code examples (to use on the new TeamMentor¹ Eclipse plug-in) so an obvious option was to use [WebGoat²](#) (whose code is currently hosted at [Google Code page³](#))

But since there wasn't a source code download option (in the current [download page⁴](#))

Filename	Summary + Labels	Uploaded	ReleaseDate	Size	DownloadCount	...
WebGoat-5.4-OWASP_Standard_Win32.zip	WebGoat 5.4 for Windows - download, unzip, click to run version <small>Featured</small>	Apr 2012	Apr 2012	76.4 MB	35252	
WebGoat-5.4.war	WebGoat 5.4 war file <small>Featured</small>	Apr 2012	Apr 2012	37.2 MB	10652	
README-5.4.txt	WebGoat 5.4 ReadMe file <small>Featured</small>	Apr 2012	Apr 2012	7.8 KB	9535	
Solving the WebGoat Labs Draft V2.pdf	Solving the WebGoat Labs	Mar 2007		106 KB	36001	

... and this project is not using Git (sorry, but I can't use SVN anymore :) ... it's too painful)

Project Home Downloads Wiki Issues **Source**

Checkout Browse Changes

Command-line access

Use this command to anonymously check out the latest project source code:

```
# Non-members may check out a read-only working copy anonymously over HTTP.
svn checkout http://webgoat.googlecode.com/svn/trunk/ webgoat-read-only
```

... I quickly created a clone of it using the `$ git svn clone -s http://webgoat.googlecode.com/svn webgoat`

... which downloaded the entire source code and available history:

¹<http://teammentor.net/>

²https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

³<https://code.google.com/p/webgoat/>

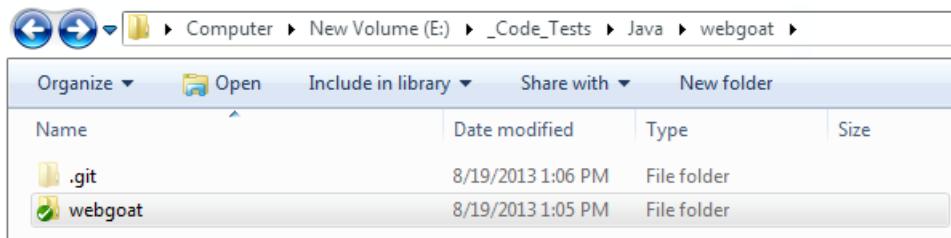
⁴<https://code.google.com/p/webgoat/downloads/list>

```
02@WIN-FGNQ5AARJ80 /e/_Code_Tests/Java
$ git svn clone -s http://webgoat.googlecode.com/svn webgoat
Initialized empty Git repository in e:/_Code_Tests/Java/webgoat/.git/
r1 = 9b0c5bf8723d7ea4739037d9b5765c053e987d69 (refs/remotes/trunk)
    A      webgoat/ExecutionException.java
    A      webgoat/Exec.java
    A      webgoat/HtmlEncoder.java
    A      webgoat/ThreadWatcher.java
    A      webgoat/ExecResults.java
r2 = 9a1ba6a4254d5b749a098018234936fe2a42a79c (refs/remotes/trunk)
    D ..   webgoat/Exec.java
```

When completed (it took a little bit since there was quite a bit of history)

```
r487 = 1c5aaffe3ccab74671b06fcfc6fd3c3731280316 (refs/remotes/webgoat-6.0)
    M      pom.xml
    M      .classpath
r488 = c433fbdb28e231123aab1201880d4a21b5dc95bb2 (refs/remotes/webgoat-6.0)
    D      .project
    D      .classpath
W: -empty_dir: branches/webgoat-6.0/.classpath
W: -empty_dir: branches/webgoat-6.0/.project
r489 = 97a05e295d0f9e3a2caca3972b9da89f89dfa8eb (refs/remotes/webgoat-6.0)
    M      scripts/webgoat_8080.bat
    M      webgoat_8080.bat
r490 = 771edfcfa6d111fe887d2c6aaeb2ef555adf01667 (refs/remotes/webgoat-6.0)
Checking out files: 100% (1206/1206), done.
Checked out HEAD:
  http://webgoat.googlecode.com/svn/trunk r480
```

I had this File Structure:



and

The screenshot shows a Windows File Explorer window with the following directory path: Computer > New Volume (E:) > _Code_Tests > Java > webgoat > webgoat. The current view is a list of files and folders within the 'webgoat' directory. The columns are Name, Date modified, Type, and Size. The files listed are:

Name	Date modified	Type	Size
doc	8/19/2013 1:05 PM	File folder	
src	8/19/2013 1:05 PM	File folder	
build.xml	8/19/2013 1:05 PM	XML Document	14 KB
pom.xml	8/19/2013 1:05 PM	XML Document	5 KB
README.txt	8/19/2013 1:05 PM	Text Document	8 KB
webgoat for SQL Server.bat	8/19/2013 1:05 PM	Windows Batch File	1 KB
webgoat.bat	8/19/2013 1:05 PM	Windows Batch File	1 KB
webgoat.sh	8/19/2013 1:05 PM	Shell Script	2 KB
webgoat_8080.bat	8/19/2013 1:05 PM	Windows Batch File	1 KB
webscarab.bat	8/19/2013 1:05 PM	Windows Batch File	1 KB

This Git repo Size:

The screenshot shows a Windows File Explorer window with the following directory path: Computer > New Volume (E:) > _Code_Tests > Java > webgoat >.git. The current view is a list of files and folders within the '.git' directory. The columns are Name, Date modified, Type, and Size. The file listed is:

Name	Date modified	Type	Size
.git	8/19/2013 1:21 PM	File folder	

A context menu is open over the '.git' folder, with the 'Properties' option selected. The 'Properties' dialog box is displayed, showing the following details for the '.git' folder:

General	Sharing	Security	Previous Versions	Customize
.git				
Type:	File folder (.git)			
Location:	E:_Code_Tests\Java\webgoat			
Size:	55.4 MB (58,129,722 bytes)			

This Git History:

File	Edit	View	Help
master	remotes	trunk	Try 3
WebGoat 5.4 distribution before WebGoat 6.0	mayhew64@gmail.com	2012-09-09 22:54:18	
alphabetized the credits	mayhew64@gmail.com	2012-09-09 22:53:44	
Changed the field1.replaceAll to field1 = field1.replaceAll. The first case does	mayhew64@gmail.com	2012-04-27 21:02:44	
Printing out full command for command[] argument. Using arrays.toList() which	mayhew64@gmail.com	2012-04-27 15:14:24	
Updated challenge screen to 5.4, added authors sections, changed ccs to su	mayhew64@gmail.com	2012-04-27 13:24:51	
Changed ubuntu name	mayhew64@gmail.com	2012-04-27 13:23:36	
Issue 42	mayhew64@gmail.com	2012-04-27 13:22:29	
Updated to 5.4	sherif.fathy@gmail.com	2012-04-26 18:29:19	
Changed errors for dropping tables to info messages	mayhew64@gmail.com	2012-04-26 18:11:32	
Fixed more /Webgoat versus /WebGoat issues.	mayhew64@gmail.com	2012-04-26 17:11:46	
Added clarification that 2 methods are needed to pass lesson	mayhew64@gmail.com	2012-04-25 19:45:10	
Added additional instructions that 2 methods are required to pass lesson	mayhew64@gmail.com	2012-04-25 19:42:20	
Allowed other criteria to solve lesson and provide some feedback for acceptat	mayhew64@gmail.com	2012-04-25 19:30:06	

which goes back all the way to 2006!

File	Edit	View	Help
Added items remotely	mayhew64 <mayhew6	2006-09-30 14:36:17	
Moved remotely	mayhew64 <mayhew6	2006-09-30 14:34:48	
git-svn-id: http://webgoat.googlecode.com/svn/trunk@11 4033779f-a91e-041c	mayhew64 <mayhew6	2006-09-30 14:30:52	
Removed file/folder	mayhew64 <mayhew6	2006-09-30 14:30:07	
git-svn-id: http://webgoat.googlecode.com/svn/trunk@9 4033779f-a91e-0410-	mayhew64 <mayhew6	2006-09-30 14:29:19	
Moved remotely	mayhew64 <mayhew6	2006-09-30 14:25:59	
Removed file/folder	mayhew64 <mayhew6	2006-09-30 14:25:36	
git-svn-id: http://webgoat.googlecode.com/svn/trunk@6 4033779f-a91e-0410-	mayhew64 <mayhew6	2006-09-30 14:12:13	
git-svn-id: http://webgoat.googlecode.com/svn/trunk@5 4033779f-a91e-0410-	mayhew64 <mayhew6	2006-09-30 14:04:08	
Removed file/folder	mayhew64 <mayhew6	2006-09-30 14:01:14	
Removed file/folder	mayhew64 <mayhew6	2006-09-30 14:00:28	
git-svn-id: http://webgoat.googlecode.com/svn/trunk@2 4033779f-a91e-0410-	mayhew64 <mayhew6	2006-09-30 13:57:26	
Initial directory structure.	(no author) <(no autho	2006-09-30 11:56:21	

These Branches:

```
$ git branch -a
* master
  remotes/tags/webgoat-5.1
  remotes/tags/webgoat-5.4
  remotes/tags/webgoat_5.4
  remotes/tags/webgoat_5.4@480
  remotes/trunk
  remotes/webgoat-6.0
  remotes/webgoat-6.0@482
```

Note that after the svn clone the current git **master** branch is the original svn **trunk**.

But as we can see by the above list, there is already an **webgoat-6.0** branch going on (in fact most of the recent code updates are done there), so here is how we can create+checkout a git tracking branch for it:

```
$ git checkout -b webgoat-6.0 remotes/webgoat-6.0
Switched to a new branch 'webgoat-6.0'

o2@WIN-FGNQ5AARJ80 /e/_Code_Tests/Java/webgoat (webgoat-6.0)
```

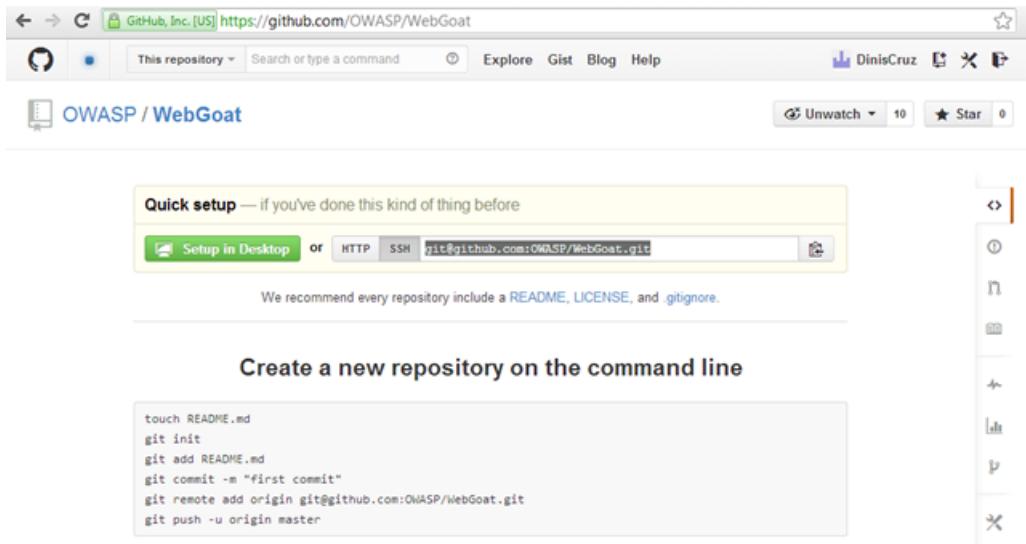
... which will make the file system look like this now:

Name	Date modified	Type	Size
.git	8/19/2013 1:16 PM	File folder	
doc	8/19/2013 1:16 PM	File folder	
java	8/19/2013 1:16 PM	File folder	
resources	8/19/2013 1:16 PM	File folder	
scripts	8/19/2013 1:16 PM	File folder	
tomcatconf	8/19/2013 1:16 PM	File folder	
webapp	8/19/2013 1:16 PM	File folder	
build.xml	8/19/2013 1:16 PM	XML Document	14 KB
pom.xml	8/19/2013 1:16 PM	XML Document	9 KB
README.txt	8/19/2013 1:16 PM	Text Document	8 KB
webgoat for SQL Server.bat	8/19/2013 1:16 PM	Windows Batch File	1 KB
webgoat.bat	8/19/2013 1:16 PM	Windows Batch File	1 KB
webgoat.sh	8/19/2013 1:16 PM	Shell Script	2 KB
webgoat_8080.bat	8/19/2013 1:16 PM	Windows Batch File	1 KB
webscarab.bat	8/19/2013 1:16 PM	Windows Batch File	1 KB

... and the Git History like this:

File	Edit	View	Help				
webgoat-6.0	remotes/webgoat-6.0	Port number typo's in the user m		EGAndrew@gmail.com	2013-05-24 21:28:46		
Messing up maven imports and not needed with maven build				mayhew64@gmail.com	2013-05-03 21:24:13		
Fixed the source code reference for the project				mayhew64@gmail.com	2012-11-20 21:35:41		
Added tomcat-maven-plugin additional src config <additionalConfigFilesDir				mayhew64@gmail.com	2012-11-20 16:56:31		
Fixed most of the maven build except the path to using tomcatconf. Moving t				mayhew64@gmail.com	2012-11-20 15:24:04		
Restructured the baseline to remove extra src/main directory structure. Add				mayhew64@gmail.com	2012-11-19 23:57:51		
Initial commit of new spring-MVC/spring security/tiles-based functionality				phillip.seay@gmail.com	2012-09-11 01:26:09		
Start of the WebGoat 6.0 project				mayhew64@gmail.com	2012-09-09 22:57:36		
remotes/tags/webgoat_5.4@480	remotes/webgoat-6.0@482	alpha!		mayhew64@gmail.com	2012-04-27 21:02:44		
Changed the field1.replaceAll to field1 = field1.replaceAll. The first case do				mayhew64@gmail.com	2012-04-27 15:14:24		
Printing out full command for command[] argument. Using arrays.toList() wr				mayhew64@gmail.com	2012-04-27 13:24:51		
Updated challenge screen to 5.4, added authors sections, changed ccs to s				mayhew64@gmail.com	2012-04-27 13:23:36		

Next step is to push this version to the newly created <https://github.com/OWASP/WebGoat⁵> repo (in OWASP GitHub organisation):



The screenshot shows a browser window with the GitHub URL <https://github.com/OWASP/WebGoat>. The page title is "OWASP / WebGoat". The main content area includes a "Quick setup" box with instructions for cloning the repository via "Setup in Desktop" or "HTTP" (selected) or "SSH". It also includes a note about including README, LICENSE, and .gitignore files. Below this is a "Create a new repository on the command line" section with a pre-filled git command:

```
touch README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin git@github.com:OWASP/WebGoat.git  
git push -u origin master
```

On the local repo add a remote:

```
o2@WIN-FGNQ5AARJ80 /e/_Code_Tests/Java/webgoat (webgoat-6.0)  
$ git remote add origin git@github.com:OWASP/WebGoat.git
```

... and *push -all*

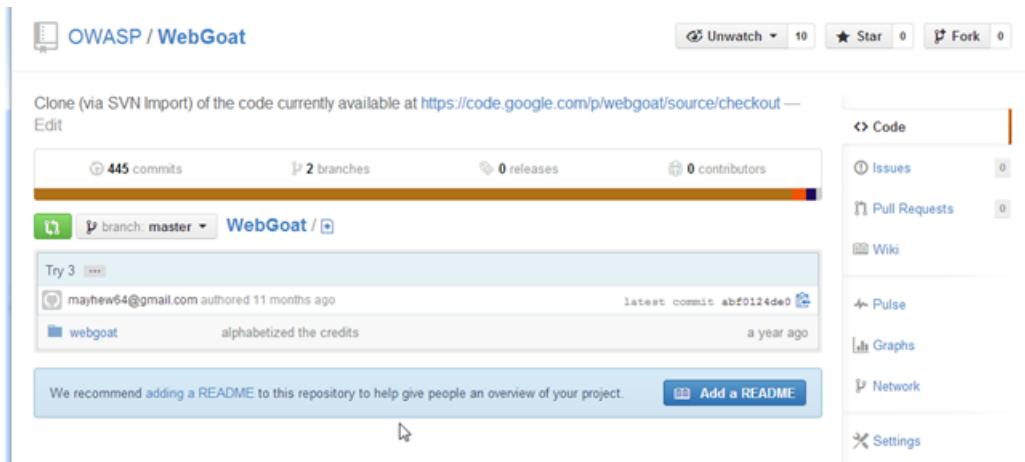
```
o2@WIN-FGNQ5AARJ80 /e/_Code_Tests/Java/webgoat (webgoat-6.0)  
$ git push --all  
Counting objects: 7450, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (5804/5804), done.  
Writing objects: 25% (1924/7450), 27.20 MiB | 1.21 MiB/s
```

Once the upload completes:

⁵<https://github.com/OWASP/WebGoat>

```
o2@WIN-FGNQSAARJ80 /e/_Code_Tests/Java/webgoat (webgoat-6.0)
$ git push --all
Counting objects: 7450, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5804/5804), done.
Writing objects: 100% (7450/7450), 44.97 MiB | 1.25 MiB/s, done.
Total 7450 (delta 3900), reused 0 (delta 0)
To git@github.com:OWASP/WebGoat.git
 * [new branch]      master -> master
 * [new branch]      webgoat-6.0 -> webgoat-6.0
```

... the code will be at GitHub:



including the **webgoat-6.0** branch:

Clone (via SVN Import) of the code currently available at <https://code.google.com/p/webgoat/source/checkout> — Edit

185 commits 2 branches 0 releases 0 contributors

branch: webgoat-6.0 WebGoat

This branch is 185 commits ahead and 445 commits behind master. Pull Request Compare

Port number typo's in the user messaging within the 8080 scripts. ...

EGAndrew@gmail.com authored 3 months ago latest commit 771edfc4d

- doc Fixed new lesson instructions location to match new file structure 4 years ago
- java Restructured the baseline to remove extra src/main directory structur... 9 months ago
- resources Restructured the baseline to remove extra src/main directory structur... 9 months ago
- scripts Port number typo's in the user messaging within the 8080 scripts. 3 months ago

Finally I updated the OWASP WebGoat page to make references to this new GitHub repo:

https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project#Downloads

Downloads

The WebGoat downloads are available at [WebGoat Google code downloads](#).
 You can sync to the current WebGoat source tree at [Google code](#).
 There is also a Git clone: available at [GitHub](#)

And that's it!

Now you can go to <https://github.com/OWASP/WebGoat>⁶ and clone (or [download the zip](#)⁷) of OWASP's WebGoat :)

⁶<https://github.com/OWASP/WebGoat>

⁷<https://github.com/OWASP/WebGoat/archive/master.zip>

10. September 2013

- Git Flow - Moving patches from one Commit into another Commit
- Example of using GitHub Pull Requests to merge changes made on Branches
- Script to Git Clone 13 repositories in order to have all TeamMentor Libraries in one folder

Git Flow - Moving patches from one Commit into another Commit

This (longish) post will cover detailed git workflows and is part of the series of blog posts that show how we use the Git Flow workflow to manage TeamMentor's source code (you will also see practical applications of GitHub's powerful of powerful features like Network Graphs and Pull Requests).

The key problem that we are going to solve, is the situation created by Michael Hidalgo¹'s TeamMentor fixes/commits/branches that were done against an commit (`38bfcd54d8046372c0ace2409324` which was originally planed to be part of the next release, but we decided that the next ***** 3.4 Release*** of TeamMentor will be based on the current 3.3.3 version (with is based on the earlier commit: `b97a470ffa173d67a9c74373593eea03eb7a2da4`).

The key reason is that he `38bfcd54d8046372c0ace2409324ecc965761504` commit(currently the parent of Michael's fixes/branches) is not stable and is going now to be the basis of the ***3.5 Release*** (this code contains a number of big changes which need more TLD and testing: native ASP.NET MVC routing, better Git support, native Markdown editor, depreciation of HTML WYSIWYG editor, and more)

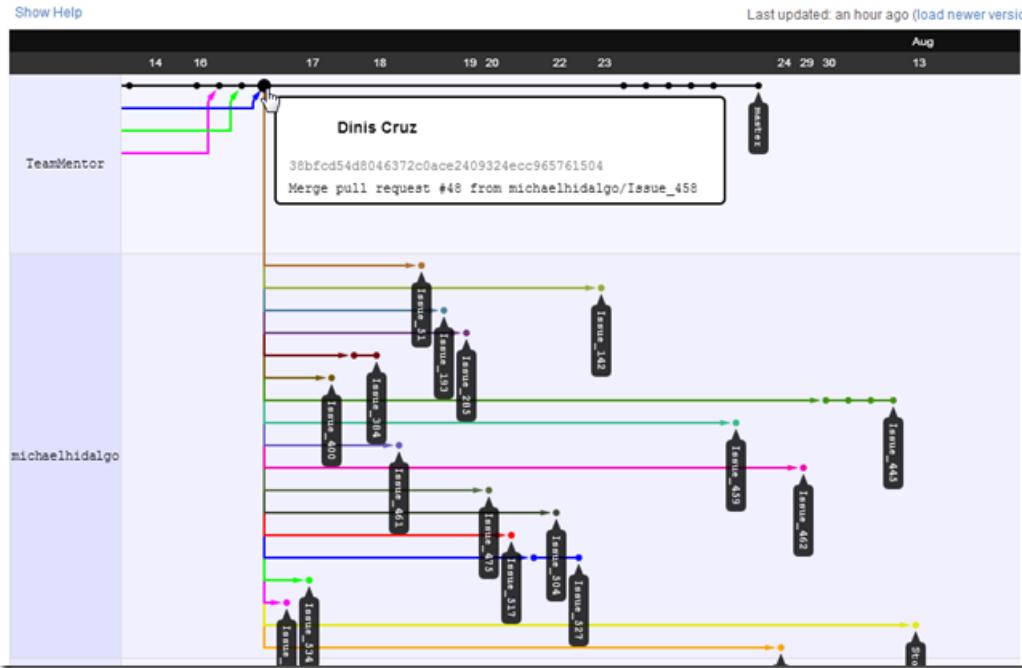
In a nutshell, we need to re-apply Michael's bug fixes to an earlier commit than the one used (i.e. backport those commits).

To start, here is what Michael's branches look like at the moment (note that all have the `38bfcd54d8046372c0ace2409324ecc965761504` commit as parent):

¹<http://blog.michaelhidalgo.info/>

The Dev network graph

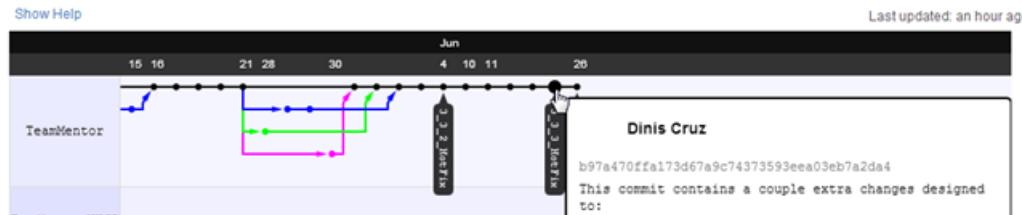
All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it works.](#)



Here is the commit (38bfcd54d8046372c0ace2409324ecc965761504) that we want to have as the parent, since this is the commit that is currently on the 3.3.3. release (and will be the basis for the 3.4 release of TeamMentor):

The Master network graph

All branches in the network using TeamMentor/Master as the reference point. [Read our blog post about how it works.](#)



Basically, what we need to do is to ‘just’ backport the branches linked to 38bfcd54d8046372c0ace2409324e commit, into the b97a470ffa173d67a9c74373593eea03eb7a2da commit

Note: since this post was getting quite long, I moved some workflows into Appendixes (included below) so that the key actions/changes can be read in sequence.

Using the workflow described in the *Appendix 1) Creating patches from Michael's branches* here are the patches to apply (i.e. these are all changes from the branches currently available in Michael's dev repository):

Name	Date modified	Type	Size
0001-Fixing-Issue-565.patch	9/4/2013 6:43 PM	PATCH File	343 KB
0001-Fixing-Issue-51.patch	9/4/2013 6:44 PM	PATCH File	3 KB
0001-Fixing-Issue_193.patch	9/4/2013 6:51 PM	PATCH File	2 KB
0001-Fixing-Issue_285.patch	9/4/2013 6:51 PM	PATCH File	347 KB
0001-Fixing-Issue_384.patch	9/4/2013 6:51 PM	PATCH File	4 KB
0002-Minor-change-on-Issue-384.patch	9/4/2013 6:51 PM	PATCH File	2 KB
0001-Fixing-Issue_400.patch	9/4/2013 6:51 PM	PATCH File	2 KB
0001-Fixing-Issue_461.patch	9/4/2013 6:52 PM	PATCH File	2 KB
0001-Fixing-Issue_475.patch	9/4/2013 6:52 PM	PATCH File	3 KB
0001-Fixing-Issue_517-In-tbot-when-setting-user-expiratio.pat...	9/4/2013 6:52 PM	PATCH File	1 KB
0001-Fixing-Issue_504.patch	9/4/2013 6:52 PM	PATCH File	340 KB
0001-Fixing-Issue_527.patch	9/4/2013 6:52 PM	PATCH File	2 KB
0002-Changing-the-welcome-message.patch	9/4/2013 6:52 PM	PATCH File	2 KB
0001-Fixing-Issue_534.patch	9/4/2013 6:52 PM	PATCH File	1,963 KB
0001-Fixing-Issue_142.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0001-Fixing-Issue_459.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0001-Fixing-Issue_462.patch	9/4/2013 6:53 PM	PATCH File	1 KB
0001-Fixing-Issue_445.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0002-Adding-a-tabulation-and-Indentation.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0003-Attempting-to-fix-indentation.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0004-Update-Right_GuidanceItem.html.patch	9/4/2013 6:53 PM	PATCH File	2 KB

After:

- Fixing the master branch and creating a feature branch for the 3.5_Release (so that **TeamMentor/Dev** master branch is in sync with **TeamMentor/Master** master branch, and the 3.5 commits are not lost)
 - ... see *Appendix 2) Creating a 3.3.3 tag and branch in Dev repository*
 - ... see *Appendix 4) Creating a 3.5_Release Feature branch***)
- Applying the 6 patches that merged without conflict
 - ... see *Appendix 3) Applying patches*

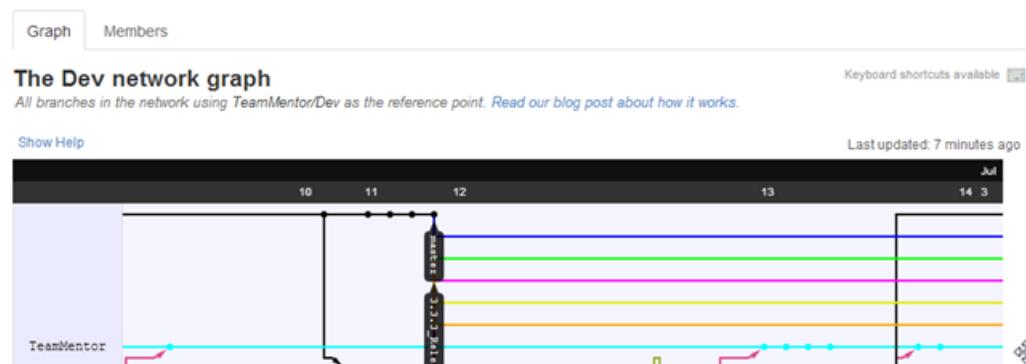
...we get the following TeamMentor/Dev 'not merged branches':

This screenshot shows the GitHub 'Branches' page for the 'TeamMentor / Dev' repository. The 'master' branch is listed as the 'Base branch'. Below it, five pull requests (Issue_459, Issue_475, Issue_400, Issue_384, and Issue_51) have been merged into master, each with a 'View pull request' and 'Compare' button. The 'Issue_142' branch is shown as being ahead of master by 1 commit. A 'Recently Active' filter is applied.

After the pull requests are made into a new 3.4_Release branch (see *Appendix 5) Creating a 3.4_Release Feature branch* for more details) we have 5 Issues/branches applied (and ready for QA):

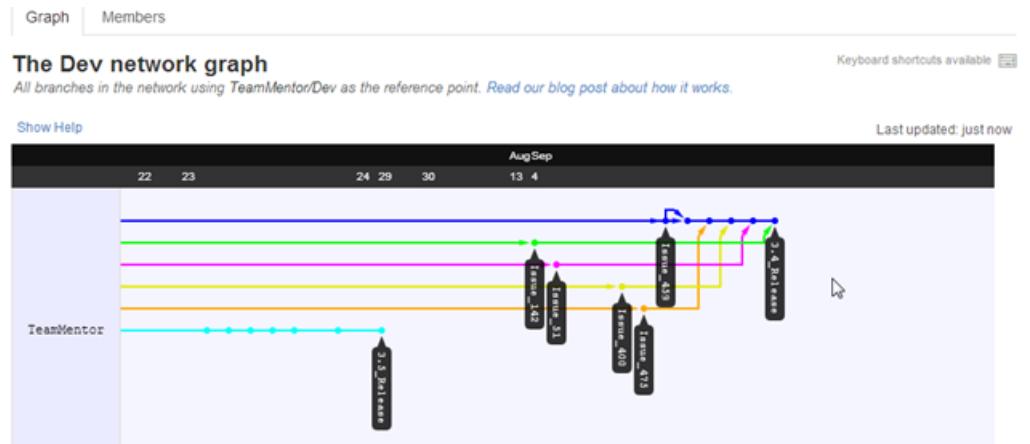
This screenshot shows the GitHub 'Branches' page for the 'TeamMentor / Dev' repository. The '3.4_Release' branch is listed as the 'Base branch'. It has 10 commits ahead of master. A 'Delete branch' and 'Compare' button are available for the 3.4_Release branch. A 'Recently Active' filter is applied.

Here is the graph view, with **TeamMentor/Dev** master (blue line below):



.... now being the parent of the **Issue_142**, **Issue_51**, **Issue_400**, **Issue_475** and **Issue_459**

branches:



This concludes the (main part of) this post, which showed how to handle the scenario where fixes (and branches) were applied to a commit whose release schedule was changed (and there was the need to back-port those changes into an earlier commit).

I think it is important to note that the workflow shown in here is a great proof of the power of Git (I can't even image doing this in SVN).

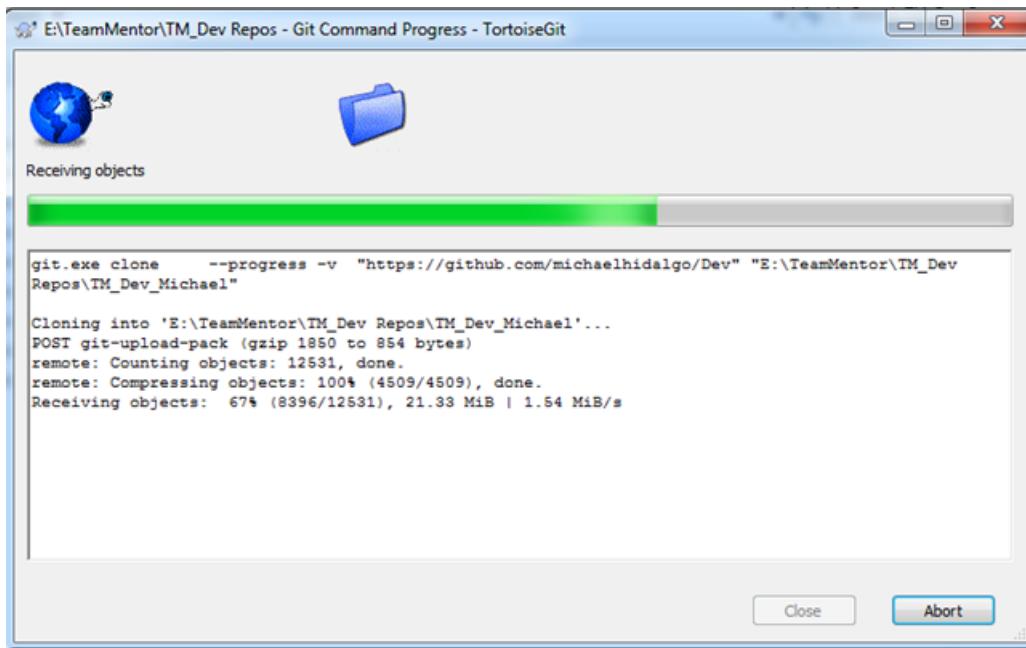
In fact, in this case, we are paying the price for not being more formal in the use of Git Flow workflows, and for not being more strategic on where we applied simple fixes (like the ones shown here).

I.e. this should be easier next time.

That said, it took me orders-of-magnitude more time to write this blog post, than to actually make these changes/fixes :)

Appendix 1) Creating patches from Michael's branches:

To create the patches, I grabbed a fresh clone of Michael's dev repo (which is a fork of TeamMentor/Dev)



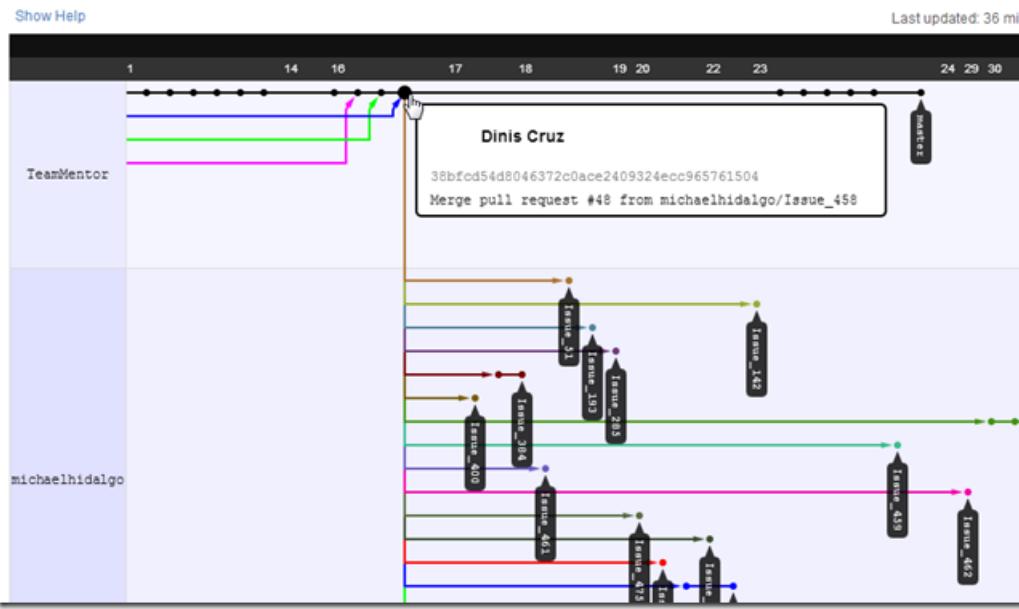
Then, on a git bash of this repository, I created a new branch that pointed to the current `38bfcd54d8046372c0ace2409324ecc965761504` commit, using the commands: `$ git checkout 38bfcd54d8046372c0ace2409324ecc965761504` `** and **$ git checkout -b Patch_Parent`

```
$ git checkout 38bfcd54d8046372c0ace2409324ecc965761504
Note: checking out '38bfcd54d8046372c0ace2409324ecc965761504'.
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:
  git checkout -b new_branch_name
HEAD is now at 38bfcd5... Merge pull request #48 from michaelhidalgo/Issue_458
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michael ((38bfcd5...))
$ git checkout -b Patch_Parent
Switched to a new branch 'Patch_Parent'
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michael (Patch_Parent)
```

The reason I picked the `38bfcd54d8046372c0ace2409324ecc965761504` commit is because this is the commit that all Michael's current branches are based on:

The Dev network graph

All branches in the network using TeamMentor/Dev as the reference point. Read our blog post about how it works.



Using the `$ git branch -a` command, we can see that this local repository/clone already contains the branches we need:

```
p2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michaelhidalgo
$ git branch -a
* 3.3.3_Release
  Issue_565
  master
  remotes/origin/Checkmarx
  remotes/origin/HEAD -> origin/master
  remotes/origin/HotFix_3.3_1
  remotes/origin/Issue_142
  remotes/origin/Issue_193
  remotes/origin/Issue_285
  remotes/origin/Issue_384
  remotes/origin/Issue_400
  remotes/origin/Issue_445
  remotes/origin/Issue_455
  remotes/origin/Issue_459
  remotes/origin/Issue_461
  remotes/origin/Issue_462
  remotes/origin/Issue_475
  remotes/origin/Issue_476
  remotes/origin/Issue_481
```

Let's start with a simple one, for example the changes on **Issue 534**²:

The screenshot shows a GitHub repository page for 'TeamMentor / Master'. The URL is <https://github.com/TeamMentor/Master/issues/534>. The issue title is 'Write better 'Welcome to TeamMentor' email message'. It is assigned to 'michaelhidalgo' and has a milestone of '3.4'. The description notes that Joe mentioned in issue #486 that the confirmation email is bad, and there's a missed opportunity to send a nice welcoming email. The text is refactored to update the template:

```
public static string EMAIL_BODY_NEW_USER_WELCOME =  
@"Hi {0}, welcome to TeamMentor.  
  
You can login with your '{1}' account at {2}  
";
```

There are 3 participants: DinisCruz, michaelhidalgo, and another user whose icon is partially visible.

whose changes are on branch **Issue_534**

²<https://github.com/TeamMentor/Master/issues/534>

gitk: TM_Repo

File Edit View Help

remotes/origin/issue_193 Fixing Issue_193
 remotes/origin/issue_51 Fixing Issue_51
 remotes/origin/issue_461 Fixing Issue_461
 remotes/origin/issue_384 Minor change on Issue 384
 Fixing Issue_384
 remotes/origin/issue_400 Fixing Issue_400
 remotes/origin/issue_534 Fixing Issue_534
 remotes/origin/issue_565 Fixing Issue_565

master remotes/origin/issue_455 remotes/origin/issue_476 remotes/origin/
 Fixing Issue_458
 Merge pull request #47 from michaelhidalgo/Issue_551
 Fixing Issue_551
 Merge pull request #45 from michaelhidalgo/Issue_407
 Fixing Issue_407

SHA1 ID: c40a80ce3650ac34e493f369a95dc30411578590 ← → Row 22 / 818

Find next prev commit containing: Search

Diff Old version New version Lines of context: 3 Ignore space change Line diff

```
index f1f5f41..fdd95e5 100644
@@ -12,11 +12,19 @@ namespace TeamMentor.CoreLib
     public static string EMAIL SUBJECT_NEW_USER_WELCOME = "Welcome to TeamMentor";
 
     public static string EMAIL_BODY_NEW_USER_WELCOME =
-@"Hi {0}, welcome to TeamMentor.
+@"Hello,
 
-You can login with your '{1}' account at {2}
+It's a pleasure to confirm that a new TeamMentor account has been created for you and that you'll
+the entire set of guidance available in the TM repository.
-
";
```

+To access the service:
+
++ Go to {0} and login at the top right-hand corner of the page.
++ Use your username : {1}.

In order to create the patch, I created a local tracking branch using the command `$ git checkout -b Issue_534 remotes/origin/Issue_534`

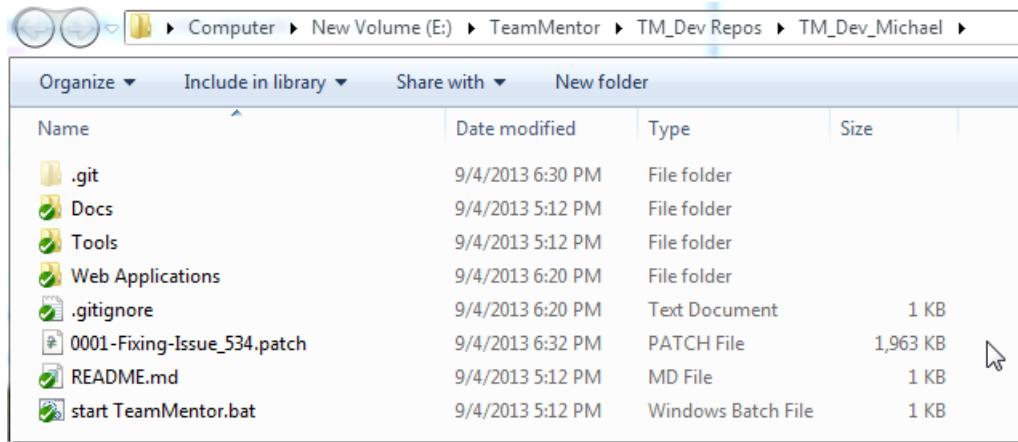
```
p2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Repo Repos/TM_Repo (Patch_Parent)
$ git checkout -b Issue_534 remotes/origin/Issue_534
Branch Issue_534 set up to track remote branch Issue_534 from origin.
Switched to a new branch 'Issue_534'

p2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Repo Repos/TM_Repo (Issue_534)
```

I then created a patch using `$ git format-patch Patch_Parent`

```
p2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev_Repos/TM_Dev_Michael (Issue_534)
$ git format-patch Patch_Parent
0001-Fixing-Issue_534.patch
```

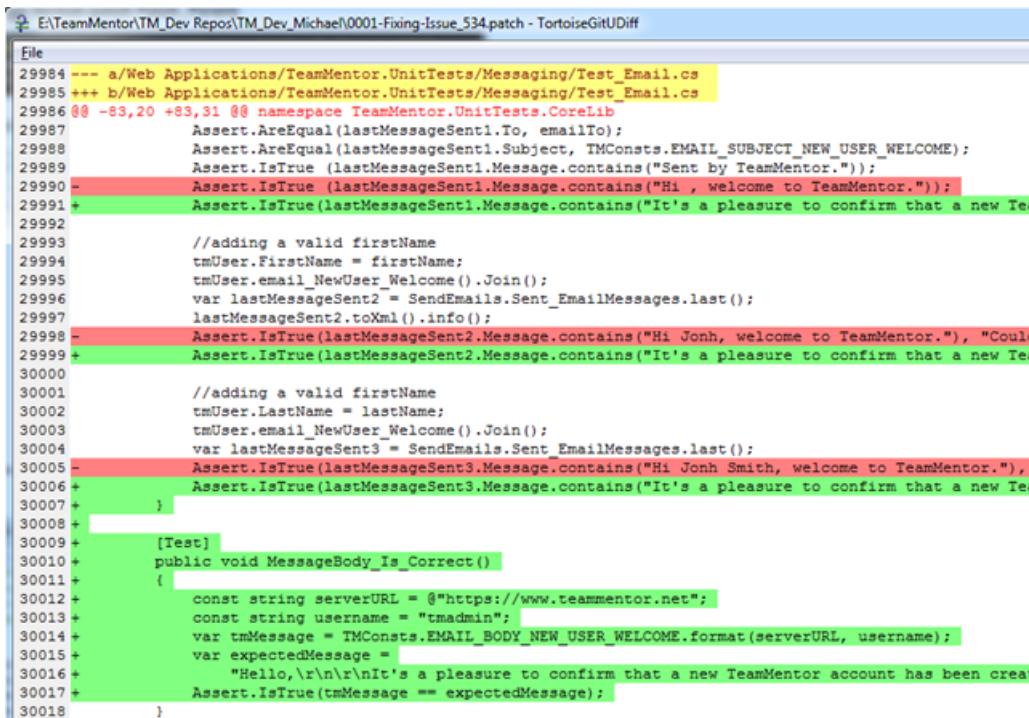
...which created the file ***0001-Fixing-Issue_534.patch***:



... containing these changes:

```
E:\TeamMentor\TM_Dev_Repos\TM_Dev_Michael\0001-Fixing-Issue_534.patch - TortoiseGitUDiff
File
1 From c40a80ce3650ac34e493f369a95dc30411578590 Mon Sep 17 00:00:00 2001
2 From: Michael <michaelfallas@gmail.com>
3 Date: Wed, 17 Jul 2013 18:25:11 -0600
4 Subject: [PATCH] Fixing Issue_534
5
6 ---
7 .../TM_Website/bin/FluentSharp.BCL.dll | Bin 1089536 -> 0 bytes
8 .../bin/O2_Platform_External_SharpDevelop.dll | Bin 1962496 -> 0 bytes
9 .../TM_Website/bin/TeamMentor.CoreLib.dll | Bin 353413 -> 300155 bytes
10 .../TM_Website/bin/TeamMentor.Website.dll | Bin 7680 -> 7680 bytes
11 .../TeamMentor.CoreLib/TM_AppCode/TMConsts.cs | 14 ++++++-----+
12 .../TM_AppCode/Utils/SendEmails.cs | 3 ++
13 .../TeamMentor.UnitTesting/Messaging/Test_Email.cs | 17 ++++++-----+
14 7 files changed, 26 insertions(+), 8 deletions(-)
15 delete mode 100644 Web Applications/TM_Website/bin/FluentSharp.BCL.dll
16 delete mode 100644 Web Applications/TM_Website/bin/O2_Platform_External_SharpDevelop.dll
```

...the these ones:



```

E:\TeamMentor\TM_Dev Repos\TM_Dev_Michael\0001-Fixing-Issue_534.patch - TortoiseGitUDiff
File
29984 --- a/Web Applications/TeamMentor.UnitTests/Messaging/Test_Email.cs
29985 +++ b/Web Applications/TeamMentor.UnitTests/Messaging/Test_Email.cs
29986 @@ -83,20 +83,31 @@ namespace TeamMentor.UnitTests.CoreLib
29987     Assert.AreEqual(lastMessageSent1.To, emailTo);
29988     Assert.AreEqual(lastMessageSent1.Subject, TMConsts.EMAIL_SUBJECT_NEW_USER_WELCOME);
29989     Assert.IsTrue(lastMessageSent1.Message.contains("Sent by TeamMentor."));
29990 -    Assert.IsTrue(lastMessageSent1.Message.contains("Hi , welcome to TeamMentor."));
29991 +    Assert.IsTrue(lastMessageSent1.Message.contains("It's a pleasure to confirm that a new Te
29992
29993     //adding a valid firstName
29994     tmUser.FirstName = firstName;
29995     tmUser.email_NewUser_Welcome().Join();
29996     var lastMessageSent2 = SendEmails.Sent_EmailMessages.last();
29997     lastMessageSent2.toXml().info();
29998 -    Assert.IsTrue(lastMessageSent2.Message.contains("Hi Jonh, welcome to TeamMentor."));
29999 +    Assert.IsTrue(lastMessageSent2.Message.contains("It's a pleasure to confirm that a new Te
30000
30001     //adding a valid firstName
30002     tmUser.LastName = lastName;
30003     tmUser.email_NewUser_Welcome().Join();
30004     var lastMessageSent3 = SendEmails.Sent_EmailMessages.last();
30005 -    Assert.IsTrue(lastMessageSent3.Message.contains("Hi Jonh Smith, welcome to TeamMentor."));
30006 +    Assert.IsTrue(lastMessageSent3.Message.contains("It's a pleasure to confirm that a new Te
30007 }
30008 +
30009+ [Test]
3010+ public void MessageBody_Is_Correct()
3011+
3012+ {
3013+     const string serverURL = @"https://www.teammentor.net";
3014+     const string username = "tmadmin";
3015+     var tmMessage = TMConsts.EMAIL_BODY_NEW_USER_WELCOME.format(serverURL, username);
3016+     var expectedMessage =
3017+         "Hello,\r\n\r\nIt's a pleasure to confirm that a new TeamMentor account has been crea
3018 }

```

Note: the reason the patch is about 1Mb is because Michael (on this branch) also committed a bunch of *.dlls which should not be there.

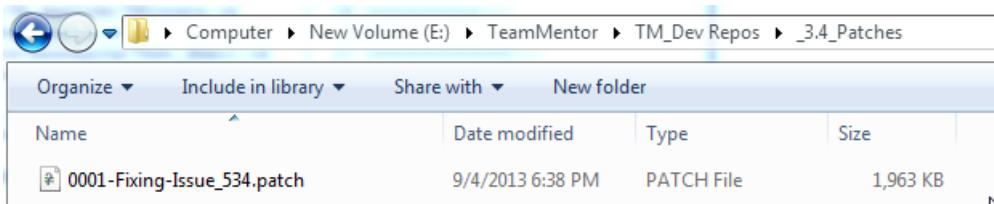
One more little thing, since we are going to create a number of these patch files, it is better to put them on a dedicated folder. This can be done using the command: `$ git format-patch Patch_Parent -o ../_3.4_Patches`

```

lo2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michael (Issue_534)
$ git format-patch Patch_Parent -o ../_3.4_Patches
../_3.4_Patches/0001-Fixing-Issue_534.patch

```

... with the '*patch file*' now being placed on the folder:

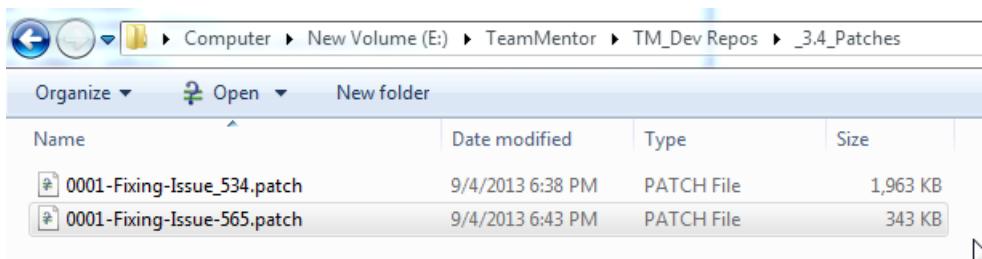


Here is the same process for Issue_565:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michael (Issue_534)
$ git checkout Issue_565
Switched to branch 'Issue_565'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michael (Issue_565)
$ git format-patch Patch_Parent -o ../../_3.4_Patches
../../_3.4_Patches/0001-Fixing-Issue-565.patch
```

... with the patch created in:



We can also create the patches without creating a tracking branch. For example here is how to create a patch for the code at the Issue_51 branch:

```
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev_Repos/TM_Dev_Michael (Issue_565)
$ git checkout remotes/origin/Issue_51
Note: checking out 'remotes/origin/Issue_51'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

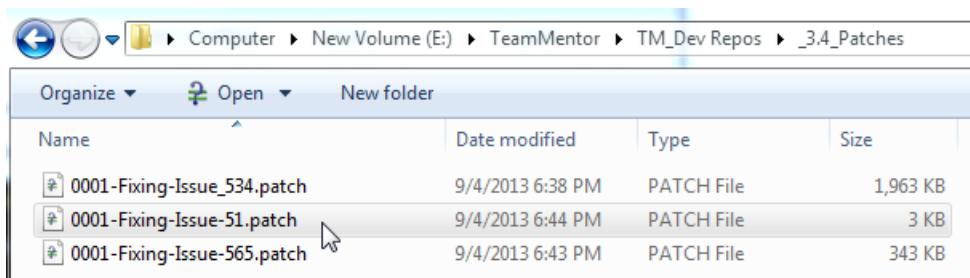
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

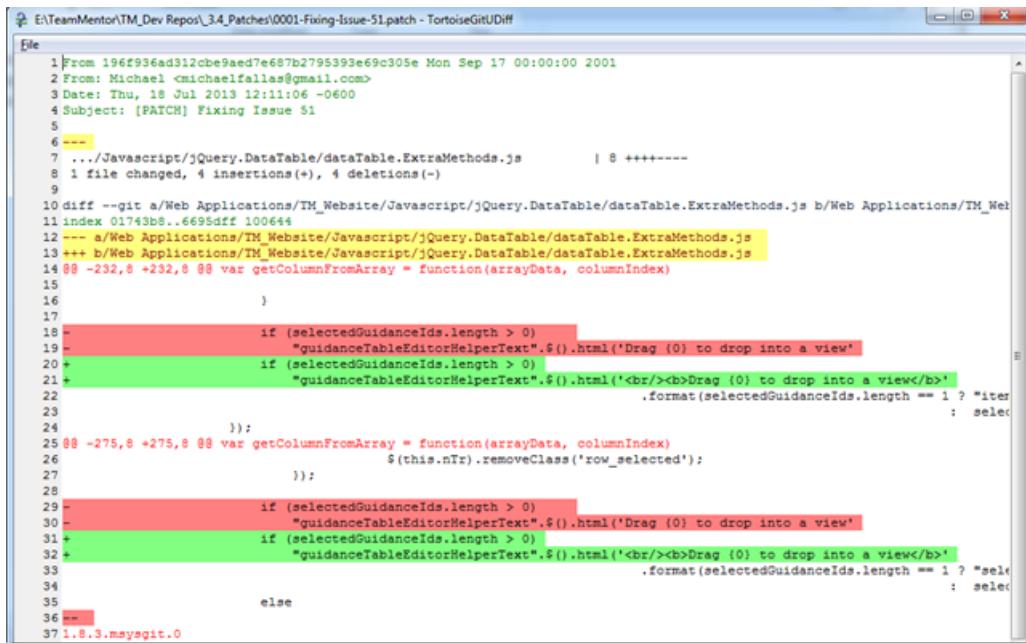
HEAD is now at 196f936... Fixing Issue 51

b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev_Repos/TM_Dev_Michael ((196f936...))
$ git format-patch Patch_Parent -o ../_3.4_Patches
../_3.4_Patches/0001-Fixing-Issue-51.patch
```

Note that the *0001-Fixing-Issue-51.patch* file is much smaller (3k) than the others



This is caused by this patch only containing text diffs (and no binaries), which is how all patches should be:



The screenshot shows a window titled "E:\TeamMentor\TM_Dev Repos\3.4_Patches\0001-Fixing-Issue-51.patch - TortoiseGitUDiff". The window displays a diff output comparing two versions of a file named "JQuery.DataTable/dataTables.ExtraMethods.js". The left side shows the original code (version 1), and the right side shows the modified code (version 2). The changes are highlighted with red and green boxes. The code includes logic for handling drag-and-drop operations on a guidance table editor.

```
1 From 196f936ad312cbe9aed7e687b2795393e69c305e Mon Sep 17 00:00:00 2001
2 From: Michael <michaelfallas@gmail.com>
3 Date: Thu, 18 Jul 2013 12:11:06 -0600
4 Subject: [PATCH] Fixing Issue 51
5
6 ===
7 .../Javascipt/jQuery.DataTable/dataTables.ExtraMethods.js | 8 ++++++-
8 1 file changed, 4 insertions(+), 4 deletions(-)
9
10 diff --git a/Web Applications/TM_Website/Javascipt/jQuery.DataTable/dataTables.ExtraMethods.js b/Web Applications/TM_Website/Javascipt/jQuery.DataTable/dataTables.ExtraMethods.js
11 index 01743b8..4695df1 100644
12 --- a/Web Applications/TM_Website/Javascipt/jQuery.DataTable/dataTables.ExtraMethods.js
13 +++ b/Web Applications/TM_Website/Javascipt/jQuery.DataTable/dataTables.ExtraMethods.js
14 @@ -232,8 +232,8 @@ var getColumnFromArray = function(arrayData, columnIndex)
15
16
17
18 -        if (selectedGuidanceIds.length > 0)
19 -            "guidanceTableEditorHelperText".$.html('Drag (0) to drop into a view')
20 +        if (selectedGuidanceIds.length > 0)
21 +            "guidanceTableEditorHelperText".$.html('<br/><b>Drag (0) to drop into a view</b>')
22
23
24    );
25 @@ -275,8 +275,8 @@ var getColumnFromArray = function(arrayData, columnIndex)
26
27
28
29 -        if (selectedGuidanceIds.length > 0)
30 -            "guidanceTableEditorHelperText".$.html('Drag (0) to drop into a view')
31 +        if (selectedGuidanceIds.length > 0)
32 +            "guidanceTableEditorHelperText".$.html('<br/><b>Drag (0) to drop into a view</b>')
33
34
35
36 ---
37 1.8.3.msysgit.0
```

Finally here are all the patch files created (containing all commits made by Michael's branches):

Name	Date modified	Type	Size
0001-Fixing-Issue-565.patch	9/4/2013 6:43 PM	PATCH File	343 KB
0001-Fixing-Issue-51.patch	9/4/2013 6:44 PM	PATCH File	3 KB
0001-Fixing-Issue_193.patch	9/4/2013 6:51 PM	PATCH File	2 KB
0001-Fixing-Issue_285.patch	9/4/2013 6:51 PM	PATCH File	347 KB
0001-Fixing-Issue_384.patch	9/4/2013 6:51 PM	PATCH File	4 KB
0002-Minor-change-on-Issue-384.patch	9/4/2013 6:51 PM	PATCH File	2 KB
0001-Fixing-Issue_400.patch	9/4/2013 6:51 PM	PATCH File	2 KB
0001-Fixing-Issue_461.patch	9/4/2013 6:52 PM	PATCH File	2 KB
0001-Fixing-Issue_475.patch	9/4/2013 6:52 PM	PATCH File	3 KB
0001-Fixing-Issue_517-In-tbot-when-setting-user-expiratio.pat...	9/4/2013 6:52 PM	PATCH File	1 KB
0001-Fixing-Issue_504.patch	9/4/2013 6:52 PM	PATCH File	340 KB
0001-Fixing-Issue_527.patch	9/4/2013 6:52 PM	PATCH File	2 KB
0002-Changing-the-welcome-message.patch	9/4/2013 6:52 PM	PATCH File	2 KB
0001-Fixing-Issue_534.patch	9/4/2013 6:52 PM	PATCH File	1,963 KB
0001-Fixing-Issue_142.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0001-Fixing-Issue_459.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0001-Fixing-Issue_462.patch	9/4/2013 6:53 PM	PATCH File	1 KB
0001-Fixing-Issue_445.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0002-Adding-a-tabulation-and-Indentation.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0003-Attempting-to-fix-indentation.patch	9/4/2013 6:53 PM	PATCH File	2 KB
0004-Update-Right_GuidanceItem.html.patch	9/4/2013 6:53 PM	PATCH File	2 KB

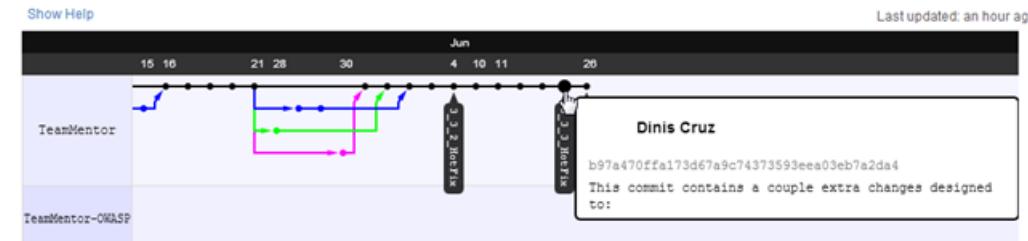
Appendix 2) Creating a 3.3.3 tag and branch in the *TeamMentor/Dev* repository

In order to be able to apply the changes into the *_TeamMentor/Master* *_master* branch, I created a branch in the current *TeamMentor/Dev* that points to the last common commit between *_TeamMentor/Master* **** and **** *_TeamMentor/Dev* (this way the commits can be pushed into *TeamMentor/Master* master branch, and eventually pulled into the *TeamMentor/Dev* 3.5_Release branch)

Since *b97a470ffa173d67a9c74373593eea03eb7a2da4* is the last commit in *TeamMentor-/Master* that also exists in *TeamMentor/Dev*, we are going to use as the parent for the patches/branches to apply:

The Master network graph

All branches in the network using TeamMentor/Master as the reference point. Read our blog post about how it works.



To do so, I started by opening up my local dev repo (currently in sync with the latest commit to Dev) , and executed `$ git checkout b97a470ffa173d67a9c74373593eea03eb7a2da4`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git checkout b97a470ffa173d67a9c74373593eea03eb7a2da4
Note: checking out 'b97a470ffa173d67a9c74373593eea03eb7a2da4'.
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:
  git checkout -b new_branch_name
HEAD is now at b97a470... This commit contains a couple extra changes designed to:
```

I then created a tracking branch (called 3.3.3_Release) and added a tag (called v3.3.3), using the commands: `$ git checkout -b 3.3.3_Release` and `$ git tag -a v3.3.3 -m '3.3.3 Release'`

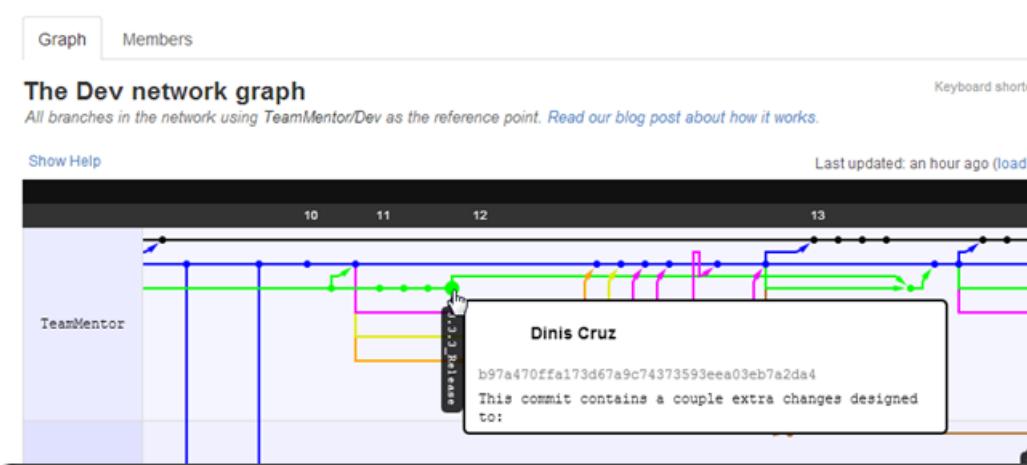
```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Michael (Issue_534)
$ git format-patch Patch_Parent -o ../../_3.4_Patches
../../_3.4_Patches/0001-Fixing-Issue_534.patch
```

I then pushed the `3.3.3_Release` branch and `v3.3.3` tag into the `TeamMentor/Dev` repository, using the commands: `$ git push dev 3.3.3_Release:3.3.3_Release` and `$ git push dev v3.3.3`

```
$ git push dev 3.3.3_Release:3.3.3_Release
Warning: Permanently added the RSA host key for IP address '192.30.252.129'
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 * [new branch] 3.3.3_Release -> 3.3.3_Release

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
$ git push dev v3.3.3
Warning: Permanently added the RSA host key for IP address '192.30.252.130'
Counting objects: 1, done.
Writing objects: 100% (1/1), 165 bytes | 0 bytes/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 * [new tag] v3.3.3 -> v3.3.3
```

Following these commands (and without the pushes that will happen next) we can see the **3.3.3_Release** tag in **TeamMentor/Dev** network graph



Appendix 3) Applying patches

We are now going to apply the patches files (previously created), into the 3.3.3_Release branch of the current local clone of **TeamMentor/Dev**

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
```

Starting with the **0001-Fixing-Issue_142.patch** which is a simple change:

```

File
1 From c96c6850f04f2d6328ddc5d59987f36adc5c20c Mon Sep 17 00:00:00 2001
2 From: Michael <michaelfallas@gmail.com>
3 Date: Tue, 23 Jul 2013 07:23:03 -0600
4 Subject: [PATCH] Fixing Issue_142
5
6 ---
7 Web Applications/TM_Website/Javascript/jQuery.jsTree/jquery.jstree.js | 2 ++
8 1 file changed, 1 insertion(+), 1 deletion(-)
9
10 diff --git a/Web Applications/TM_Website/Javascript/jQuery.jsTree/jquery.jstree.js b/Web Applications/TM_Website/Javascript/jQuery.jsTree/jquery.jstree.js
11 index 95797f3..602a4cb 100644
12 --- a/Web Applications/TM_Website/Javascript/jQuery.jsTree/jquery.jstree.js
13 +++ b/Web Applications/TM_Website/Javascript/jQuery.jsTree/jquery.jstree.js
14 @@ -1800,7 +1800,7 @@
15         d = $("<li />");
16         if(js.attr) { d.attr(js.attr); }
17         if(js.metadata) { d.data(js.metadata); }
18 -        if(js.state) { d.addClass("jstree-" + js.state); }
19 +        if(js.state) { d.addClass("jstree-last jstree-leaf"); }
20         if(!$.isArray(js.data)) { tmp = js.data; js.data = []; js.data.push(tmp); }
21         $.each(js.data, function (i, m) {
22             tmp = $("<a />");
23     -->
24 1.8.3.msysgit.0

```

To get a preview of what will change when we apply a patch, we can use the command: \$ *git apply --stat ../_3.4_Patches/0001-Fixing-Issue_142.patch*

```

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
$ git apply --stat ../_3.4_Patches/0001-Fixing-Issue_142.patch
.../Javascript/jQuery.jsTree/jquery.jstree.js | 2 ++
1 file changed, 1 insertion(+), 1 deletion(-)

```

To see if we are going to have any errors when applying a patch, we can use the command: \$ *git apply --check ../_3.4_Patches/0001-Fixing-Issue_142.patch*

```

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
$ git apply --check ../_3.4_Patches/0001-Fixing-Issue_142.patch
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)

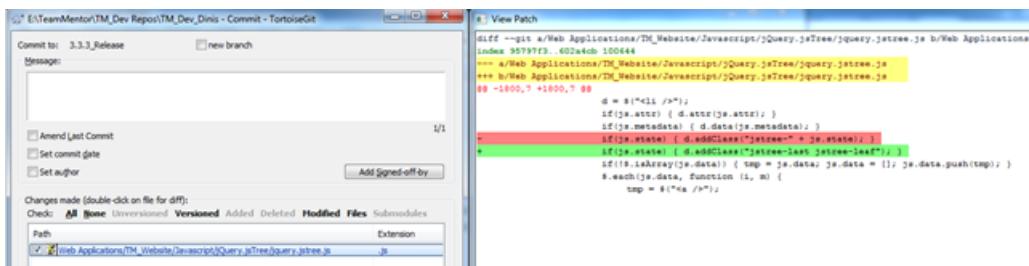
```

In this case, the fact that we saw no messages on the *--check* command (shown above), means that we can merge this patch file ok:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
$ git apply ../../3.4_Patches/0001-Fixing-Issue_142.patch

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
$ git status
# On branch 3.3.3_Release
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   Web Applications/TM_Website/Javascript/jquery.jsTree/jquery.jstree.js
no changes added to commit (use "git add" and/or "git commit -a")
```

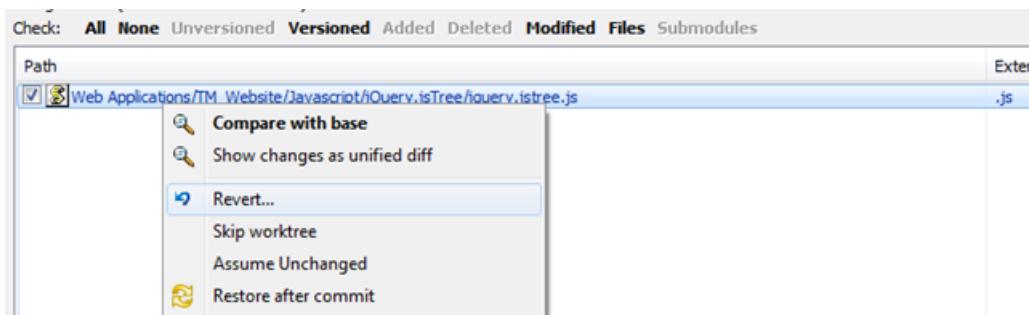
... in this case the change was applied on top of our current branch code (with no commit added)



But that has the problem that there was no commit made (just the files changed on disk).

Since we want to preserve the original commit we, will need to can use another command.

First lets reset the current change:



... and before we apply the 0001-Fixing-Issue_142.patch, lets create the Issue_142 branch, using the command `git checkout -b Issue_142`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
$ git checkout -b Issue_142
Switched to a new branch 'Issue_142'

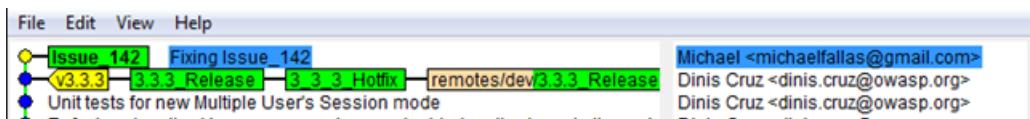
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_142)
```

Now lets apply the patch this using the command: `$ git am --signoff<../_3.4_Patches/0001-Fixing-Issue_142.patch`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_142)
$ git am --signoff <../_3.4_Patches/0001-Fixing-Issue_142.patch
Applying: Fixing Issue_142

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_142)
```

...which will add a commit containing the original commit message and author:



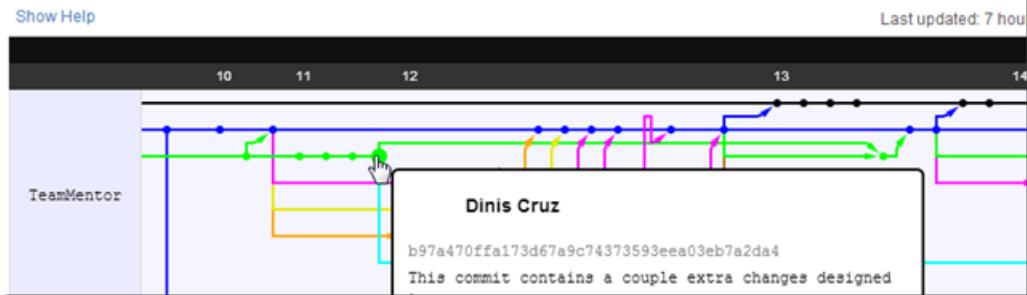
Next we push this branch into TeamMentor/Dev

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_142)
$ git push dev Issue_142:Issue_142
Warning: Permanently added the RSA host key for IP address '192.30.252.131'
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 615 bytes | 0 bytes/s, done.
Total 7 (delta 6), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 * [new branch]      Issue_142 -> Issue_142
```

And confirm that the Issue_142 changes are in the correct location (i.e with the `b97a470ffa173d67a9c743` commit as its parent):

The Dev network graph

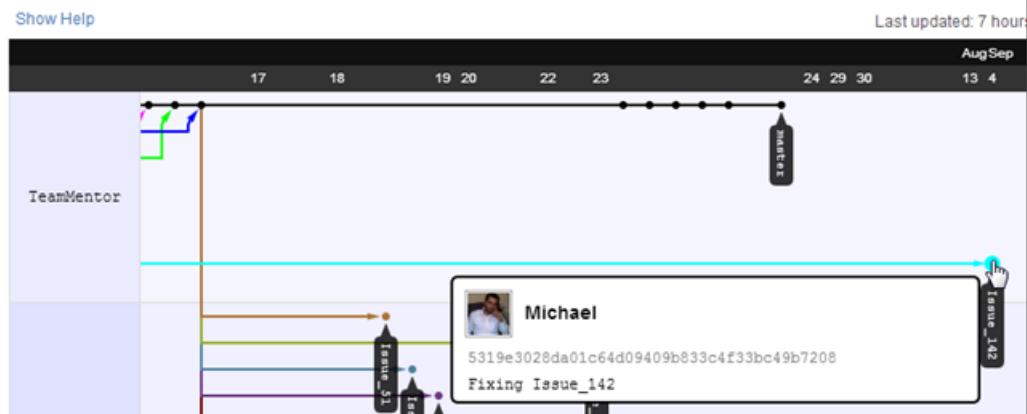
All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it works.](#)



Note how the light blue line is connected from the `b97a470ffa173d67a9c74373593eea03eb7a2da4` commit (see above) into the newly pushed `5319e3028da01c64d09409b833c4f33bc49b7208` commit (see below)

The Dev network graph

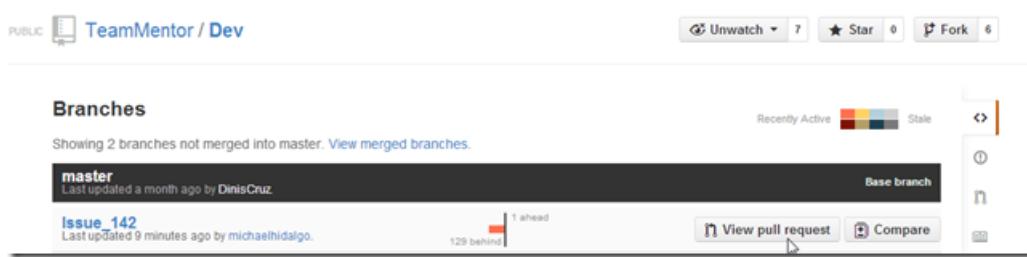
All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it works.](#)



... which is the current head of the Issue_142 branch

```
o2@WIN-FGNQ5AARJ8O /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_142)
$ git rev-parse HEAD
5319e3028da01c64d09409b833c4f33bc49b7208
```

The next image shows how we can use GitHub's UI to create/view the pull request for this branch:

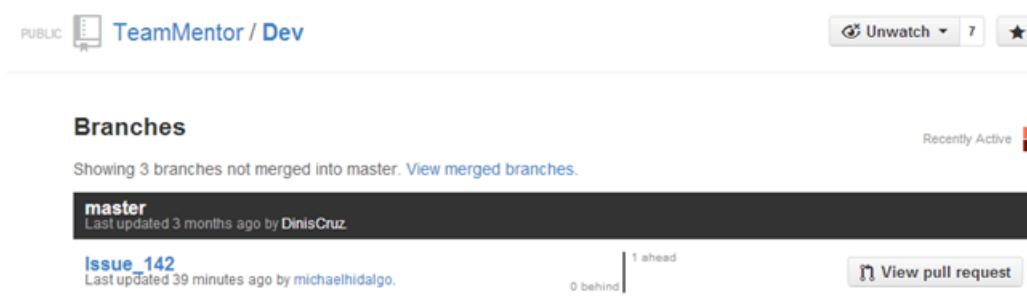


Note how in the screenshot above the ***Issue_142*** branch is 129x commit behind master.

That is caused by the fact that master is currently at the commit **16354b3ec1757f56f0ee1594de3c72bb506fe** and it should be at the commit **b97a470ffa173d67a9c74373593eea03eb7a2da4**

See ***Appendix 5) Creating a 3.4_Release Feature branch and merging branches*** for how that was fixed.

After mapping the current master commit into a new the 3.5_Release branch and doing a force reset to the master branch, we get the ***Issue_142*** branch correctly set-up with 1x commits ahead and 0x commits behind the master branch:



With TeamMentor/Dev master branch in the correct location, lets apply more patches into it:

For example Issue 51, using the commands:

```
$ git apply -check ../_3.4_Patches/0001-Fixing-Issue-51.patch (check if patch can be applied)
$ git checkout -b Issue_51 (create patch branch)
$ git am -signoff < ../_3.4_Patches/0001-Fixing-Issue-51.patch (apply patch and preserve original commit)
$ git push dev Issue_51:Issue_51 (push branch into GitHub)
```

```

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git apply --check ../_3.4_Patches/0001-Fixing-Issue-51.patch

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git checkout -b Issue_51
Switched to a new branch 'Issue_51'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_51)
$ git am --signoff < ../_3.4_Patches/0001-Fixing-Issue-51.patch
Applying: Fixing Issue 51

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_51)
$ git push dev Issue_51:Issue_51
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 646 bytes | 0 bytes/s, done.
Total 7 (delta 6), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 * [new branch]      Issue_51 -> Issue_51

```

This makes the **Issue_51** branch to also be 1x ahead and 0x behind commits of the master branch:

The screenshot shows a GitHub repository interface. At the top, there's a header with 'PUBLIC' and the repository name 'TeamMentor / Dev'. To the right are 'Unwatch', a dropdown menu, and a star icon. Below the header, the word 'Branches' is highlighted in blue. A note says 'Showing 4 branches not merged into master. View merged branches.' On the right, there's a 'Recently Active' button. The branches listed are:

- master**: Last updated 3 months ago by DinisCruz.
- Issue_51**: Last updated 2 minutes ago by michaelhidalgo. Status: 1 ahead, 0 behind. Has a 'View pull request' button.
- Issue_142**: Last updated an hour ago by michaelhidalgo. Status: 1 ahead, 0 behind. Has a 'View pull request' button.

With this workflow in place, I quickly did the same workflow for the branches: **Issue_384**, **Issue_400**, **Issue_475** and **Issue_459**

At the moment we have these branches to merge (*Appendix 5) Creating a 3.4_Release Feature branch and merging branches* will show them in action):

The screenshot shows a GitHub repository named 'TeamMentor / Dev'. The 'Branches' page is displayed, showing 8 branches not merged into master. The master branch is the base branch. The branches listed are:

- Issue_459**: Last updated a minute ago by michaelhidalgo. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_475**: Last updated 10 minutes ago by michaelhidalgo. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_400**: Last updated 13 minutes ago by michaelhidalgo. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_384**: Last updated 16 minutes ago by michaelhidalgo. Status: 2 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_51**: Last updated 26 minutes ago by michaelhidalgo. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_142**: Last updated an hour ago by michaelhidalgo. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.

Note that there were numerous patches (534, 565, 193, 285, 461, 517, 504, 527, 462 and 445) that didn't merge correctly.

For example this is what happened for the **0001-Fixing-Issue-565.patch** when executing the command `$ git apply --check ../../_3.4_Patches/0001-Fixing-Issue-565.patch`

```
o2@WIN-FGNQ5ARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.3.3_Release)
$ git apply --check ../../_3.4_Patches/0001-Fixing-Issue-565.patch
error: patch failed: Web Applications/TM_Website/TBot/Views/User_Edit.cshtml:73
error: Web Applications/TM_Website/TBot/Views/User_Edit.cshtml: patch does not apply
error: the patch applies to 'Web Applications/TM_Website/bin/TeamMentor.CoreLib.dll' (cannot match the current contents).
error: Web Applications/TM_Website/bin/TeamMentor.CoreLib.dll: patch does not apply
error: Web Applications/TM_Website/bin/TeamMentor.Website.dll: No such file or directory
error: patch failed: Web Applications/TeamMentor.CoreLib/TM_AppCode/Schemas/Models/TM_Us
error: Web Applications/TeamMentor.CoreLib/TM_AppCode/Schemas/Models/TM_User.cs: patch d
```

These will need to be handled separately (which is a topic for another blog post, since this one is already getting a bit long :))

Appendix 4) Creating a 3.5_Release Feature branch

In order to make the current **TeamMentor/Dev** match the **TeamMentor/Master** in terms of the master branch, we need to move the current master of **TeamMentor/Dev** into a feature branch called **3.5_Release** (in a way we were using the master of **TeamMentor/Dev** as a 'feature branch' which was ok if that code was going to become the 3.4 release (which now it isn't).

First step is to move into the current master using the command `$ git checkout master`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (Issue_142)
$ git checkout master
Switched to branch 'master'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
```

Then we create the 3.5_Release feature branch using the command `$ git checkout -b 3.5_Release`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git checkout -b 3.5_Release
Switched to a new branch '3.5_Release'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.5_Release)
```

Next we push this branch into *TeamMentor/Dev*

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.5_Release)
$ git push dev 3.5_Release:3.5_Release
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 751 bytes | 0 bytes/s, done.
Total 7 (delta 6), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 * [new branch] 3.5_Release -> 3.5_Release

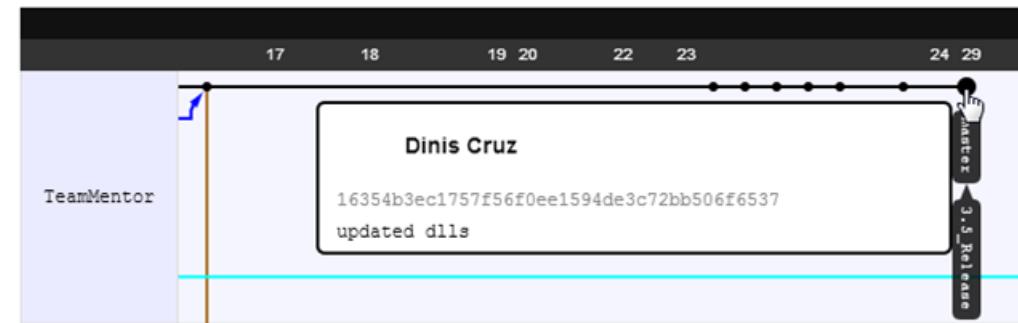
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.5_Release)
```

At this moment, in the GitHub repo, TeamMentor/Dev's master and 3.5_Release point to the same commit (16354b3ec1757f56f0ee1594de3c72bb506f6537):

The Dev network graph

All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it works](#)

Show Help

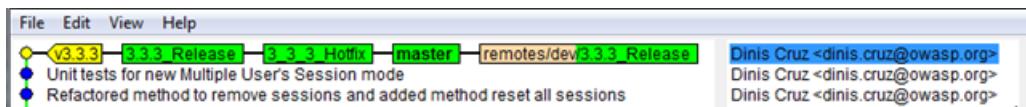


Now comes the sledgehammer :)

We're going to (first locally) do a hard reset into the `b97a470ffa173d67a9c74373593eea03eb7a2da4` commit, using the command `$ git reset --hard b97a470ffa173d67a9c74373593eea03eb7a2da4` (remember that this commit is the common one between *TeamMentor/Master* and *TeamMentor/Dev*)

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev_Repos/TM_Dev_Dinis (master)
$ git reset --hard b97a470ffa173d67a9c74373593eea03eb7a2da4
HEAD is now at b97a470 This commit contains a couple extra changes designed to:
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev_Repos/TM_Dev_Dinis (master)
```

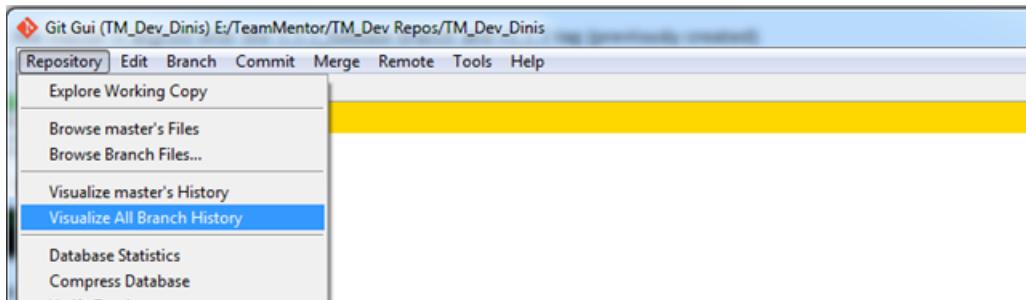
After this hard reset, the *TeamMentor/Dev* master is aligned with the `3.3.3_Release` branch and `v3.3.3` tag (previously created)



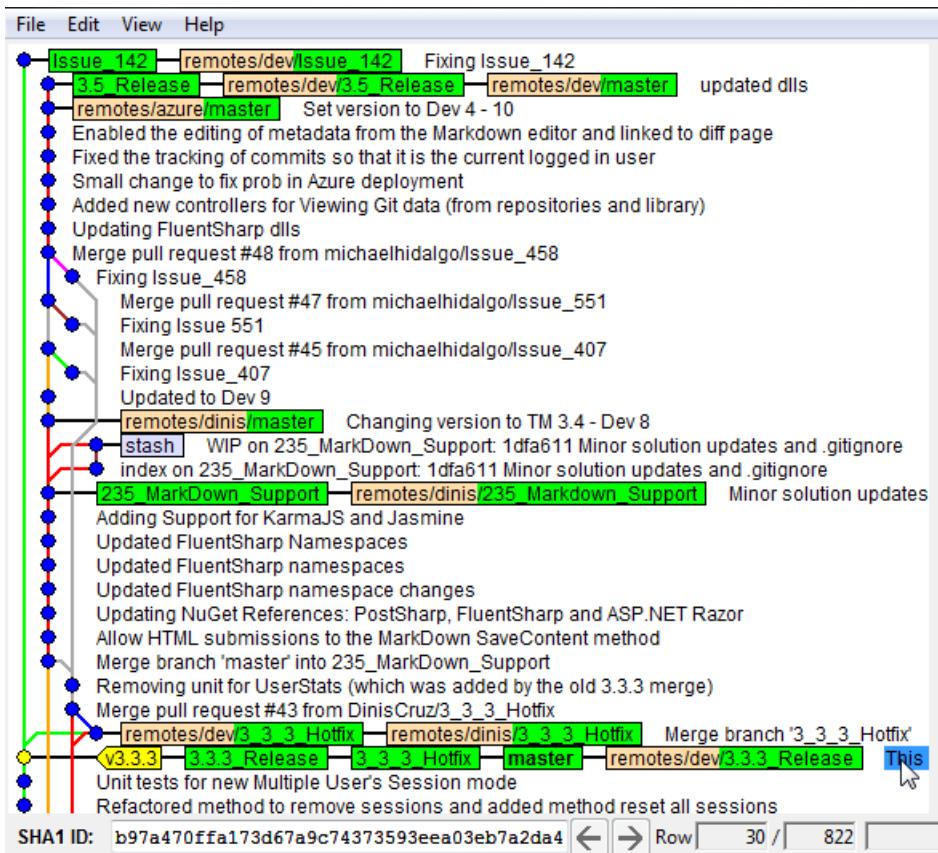
We can also double check this, by using the command `$ git gui`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev_Repos/TM_Dev_Dinis (master)
$ git gui
```

... followed by the *Visualize all Branch History* menu option:



...and see that the `Issue_142` branch is now a child of the current *TeamMentor/Dev* master (which is in sync with the *TeamMentor/Master* master)



Finally we are ready to apply the sledgehammer to the repository hosted at GitHub, by forcing a push using the command `$ git push -f dev master:master`

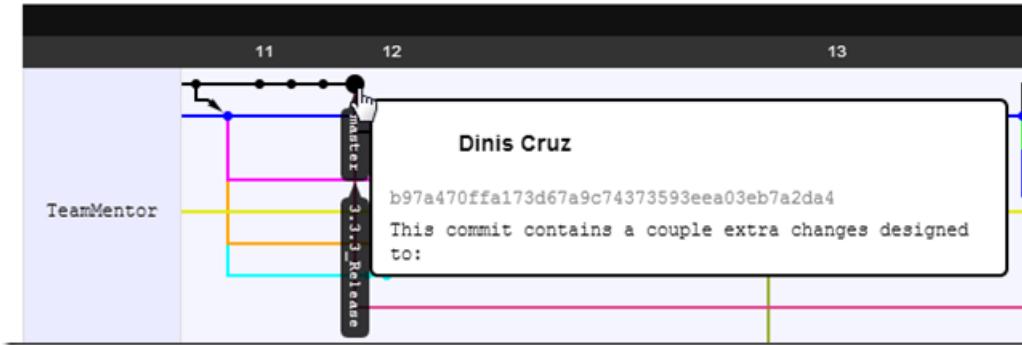
```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git push -f dev master:master
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 + 16354b3...b97a470 master -> master (forced update)
```

Which makes the TeamMentor/Master look like this:

The Dev network graph

All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it works.](#)

Show Help



... with the **Issue_142** branch having the master/3.3.3_Release branch as parent (see rouge/brown line)

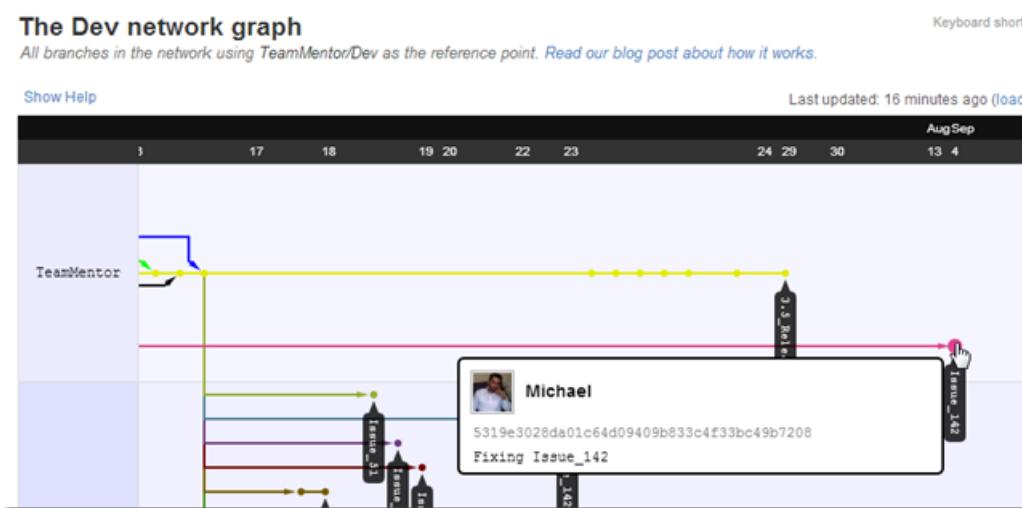
The Dev network graph

All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it works.](#)

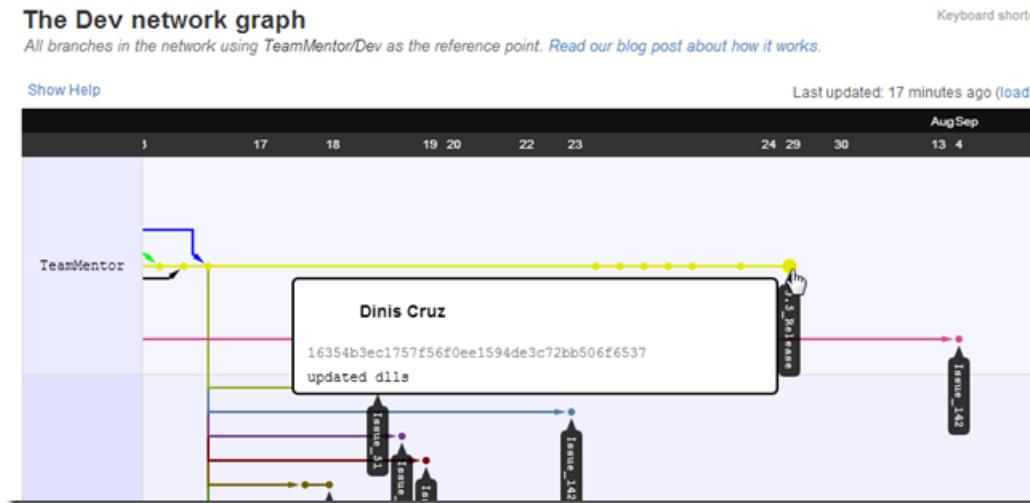
Keyboard short

Show Help

Last updated: 16 minutes ago ([load](#))



... and the 3.5_Release branch containing the commits that were previously in the master branch (see yellow line)



Finally a look at the current branches in **TeamMentor/Dev** shows that the **Issue_142** is correctly 1x commit ahead and 0x behind the master branch (which means that it is ready for a pull request)

PUBLIC TeamMentor / Dev Unwatch 7

Recently Active

Branches

Showing 3 branches not merged into master. [View merged branches.](#)

master	Last updated 3 months ago by DinisCruz .
Issue_142	Last updated 39 minutes ago by michaelhidalgo . 1 ahead 0 behind View pull request

Appendix 5) Creating a 3.4_Release Feature branch and merging branches

At this point we have these branches ready to commit (via a pull request)

Branches

Showing 8 branches not merged into master. [View merged branches](#).

Branch	Last updated	Author	Status	Action Buttons
master	3 months ago	DinisCruz	Base branch	
Issue_459	9 minutes ago	michaelhidalgo	1 ahead 0 behind	View pull request Compare
Issue_475	18 minutes ago	michaelhidalgo	1 ahead 0 behind	View pull request Compare
Issue_400	21 minutes ago	michaelhidalgo	1 ahead 0 behind	View pull request Compare
Issue_384	24 minutes ago	michaelhidalgo	2 ahead 0 behind	View pull request Compare
Issue_51	34 minutes ago	michaelhidalgo	1 ahead 0 behind	View pull request Compare
Issue_142	an hour ago	michaelhidalgo	1 ahead 0 behind	View pull request Compare
3.5_Release	a month ago	DinisCruz	130 ahead 0 behind	Delete branch Compare

Instead of merging them into the **TeamMentor/Dev** master branch, we are going to create a **TeamMentor/Dev 3.4_Release** branch using the command `$ git checkout -b 3.4_Release` and push it to TeamMentor/Dev using the command `$ git push dev 3.4_Release:3.4_Release`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (master)
$ git checkout -b 3.4_Release
Switched to a new branch '3.4_Release'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_Dev_Dinis (3.4_Release)
$ git push dev 3.4_Release:3.4_Release
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 * [new branch]      3.4_Release -> 3.4_Release
```

The reason for this branch is so that **TeamMentor/Dev** master branch is aligned with **TeamMentor/Master** master branch (which is the current official release), and only QA'd changes are pushed into **TeamMentor/Master** (first into 3.4_Release branch, and eventually into the official **TeamMentor/Master** master branch (note that we will most likely rename the **TeamMentor/Master** repo into **TeamMentor/Release**)

Next step is to create a pull request from the current Issue_XYZ branches into the 3.4_Release branch.

Let's start with **Issue_459**, by clicking on its Compare button:

Showing 8 branches not merged into master. View merged branches.

The screenshot shows a list of branches in a GitHub repository. At the top, it says "Showing 8 branches not merged into master. View merged branches." Below this, there is a table with three rows:

- master**: Last updated 3 months ago by DinisCruz. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_459**: Last updated 15 minutes ago by michaelhidalgo. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_475**: Last updated 23 minutes ago by michaelhidalgo. Status: 1 ahead, 0 behind. Buttons: View pull request, Compare.
- Issue_400**: Status: 1 ahead. Buttons: View pull request, Compare.

On the next page, click on Edit:

The screenshot shows the same GitHub repository page as above, but with a different focus. The "Edit" button in the top right corner of the master branch row is highlighted with a cursor, indicating where to click to change the base branch.

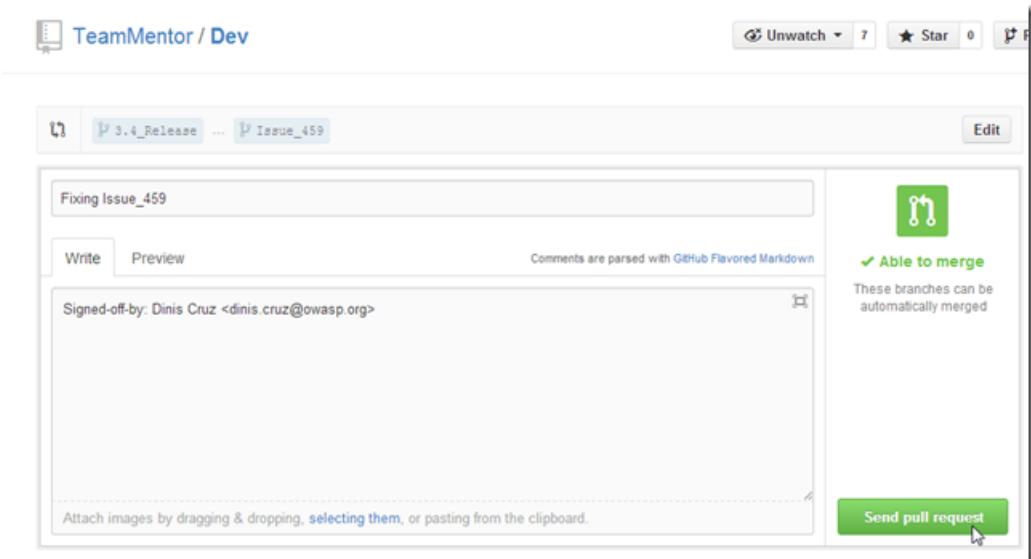
... to change the base branch (into 3.4_Release):

The screenshot shows the "Edit" screen for the master branch. In the top left, it says "base: master ... compare: Issue_459". Below this, a "Choose a base branch" dropdown is open, showing a list of branches. The "3.4_Release" branch is selected and highlighted with a blue background and a checked checkbox icon. Other options in the list include "3.3.3_Release", "3.5_Release", and "Issue_51".

Then click on the *Click to create a pull request for this comparison* link



... click on the *Send the Pull Request* button:



... click on the **Merge pull request** button

DinisCruz wants to merge 1 commit into `3.4_Release` from `Issue_459`

Fixing Issue_459

No one is assigned No milestone

Signed-off-by: Dinis Cruz dinis.cruz@owasp.org

2 participants

`michaelhidalgo` added a commit a month ago
`michaelhidalgo` Fixing Issue_459 ... f020358

You can add more commits to this pull request by pushing to the `Issue_459` branch on TeamMentor/Dev

This pull request can be automatically merged.
 You can also merge branches on the command line.

Merge pull request

... and the *Confirm Merge* button:

Merge pull request #68 from TeamMentor/Issue_459

Fixing Issue_459

DinisCruz
 dinis.cruz@owasp.org

Confirm merge

We could now delete the branch (but I'm not going to do that at this stage, since first I want to see these merged branches in a GitHub Network Graph):

Pull request successfully merged and closed
 You're all set—the `Issue_459` branch can be safely deleted.

Delete branch

Back into the **Branches not merged into master** list, although the `Issue_459` branch is still 1x ahead of master, we now have the `3.4_Release` branch with 2x commits ahead:

Branches

Showing 9 branches not merged into master. [View merged branches.](#)

Branch	Last updated	Author	Status	Actions
master	3 months ago	DinisCruz	Base branch	
3.4_Release	just now	DinisCruz	2 ahead 0 behind	Delete branch Compare
Issue_459	20 minutes ago	michaelhidalgo	1 ahead 0 behind	View pull request Compare
Issue_475	29 minutes ago	michaelhidalgo	1 ahead 0 behind	View pull request Compare

The two commits of the **3.4_Release** branch are one from the **Issue_459** branch and one from the pull request merge (note above how we could now do a Pull request from this **3.4_Release** branch into the **master** branch):

TeamMentor / Dev

master ... 3.4_Release

Click to create a pull request for this comparison

2 commits 1 file changed 0 comments 2 contributors

Sep 04, 2013

michaelhid... Fixing Issue_459 ... f020358

DinisCruz Merge pull request #68 from TeamMentor/Issue_459 ... b28adcc

Show Diff Stats

After doing the same workflow for **Issue_475** branch:

DinisCruz opened this pull request just now
Fixing Issue_475

No one is assigned ⚙ No milestone ⚙

Signed-off-by: Dinis Cruz dinis.cruz@owasp.org

2 participants

michaelhidalgo added a commit 2 months ago
michaelhid... Fixing Issue_475 ... 70fe32c

DinisCruz referenced this pull request from a commit just now
DinisCruz Merge pull request #69 from TeamMentor/Issue_475 ... d9f4f80

Merged DinisCruz merged commit d9f4f80 into **3_4_Release** from **Issue_475** just now

Closed DinisCruz closed the pull request just now

Pull request successfully merged and closed
 You're all set—the **Issue_475** branch can be safely deleted.

... the **3_4_Release** branch is 4 commits ahead:

Branches

Showing 9 branches not merged into master. [View merged branches.](#)

master Last updated 3 months ago by DinisCruz **Base branch**

3.4_Release Last updated a minute ago by DinisCruz. 4 ahead | 0 behind

And after doing the same workflow for the **Issue_142**, **Issue_51** and **Issue_400** branches/issues, the **3_4_Release** is 10 commits ahead (with 5 Issues_Xyz applied):

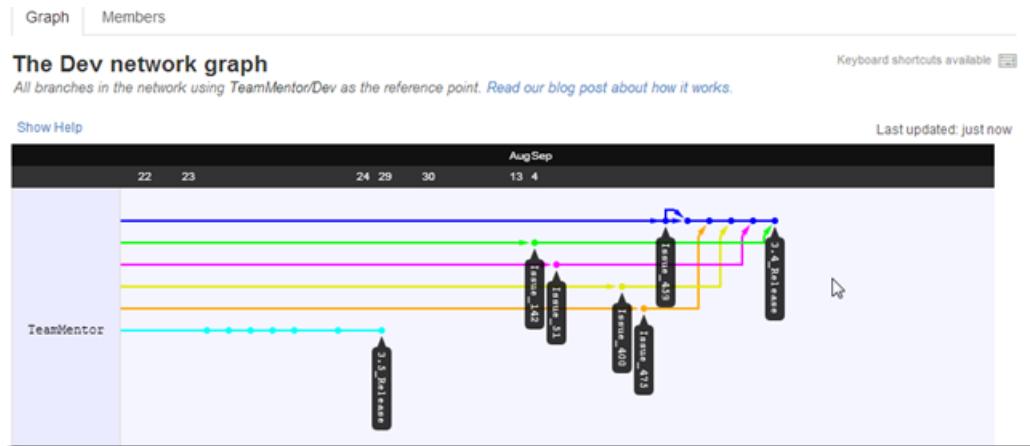
Branches

Showing 9 branches not merged into master. [View merged branches.](#)

master Last updated 3 months ago by DinisCruz **Base branch**

3.4_Release Last updated just now by DinisCruz. 10 ahead | 0 behind

The TeamMentor/Dev graph also shows this workflow in action (note that If I had deleted branches after the pull request, we wouldn't see the tags in this network graph)



One important note is that the ***Issue_384*** didn't merge automatically with the ***3.4_Release***, which means that there is a conflict between one of the changes made by the applied branches and this code (i.e. Michael will need to fix this and resubmit the patch)

Wrapping up: Feedback and better git commands:

If you made it this far to the end, it would be great to have some feedback on this git workflow

(and solution).

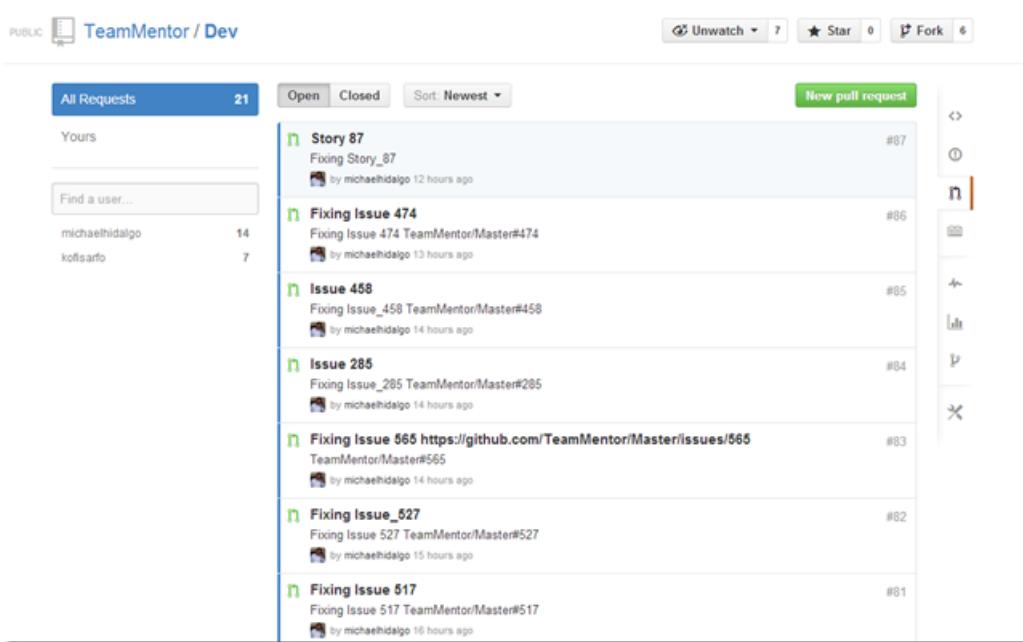
And if you know of better ways to do solve probs like this one, please ping us with your ideas, since there is still far too much Git functionality that I/we are not aware of.

Example of using GitHub Pull Requests to merge changes made on Branches

After the fixes explained in the [Git Flow - Moving patches from one Commit into another Commit³](#) post and the reset of the TeamMentor 3.4 branch, Michael reapplied his other changes/fixes to the correct 3.4 commit, and I'm now in the process merging his Pull Requests into the 3.4_Release branch (and eventually into the master branch).

This post walks through my current workflow.

At the moment there are a number of Pull Requests to process:

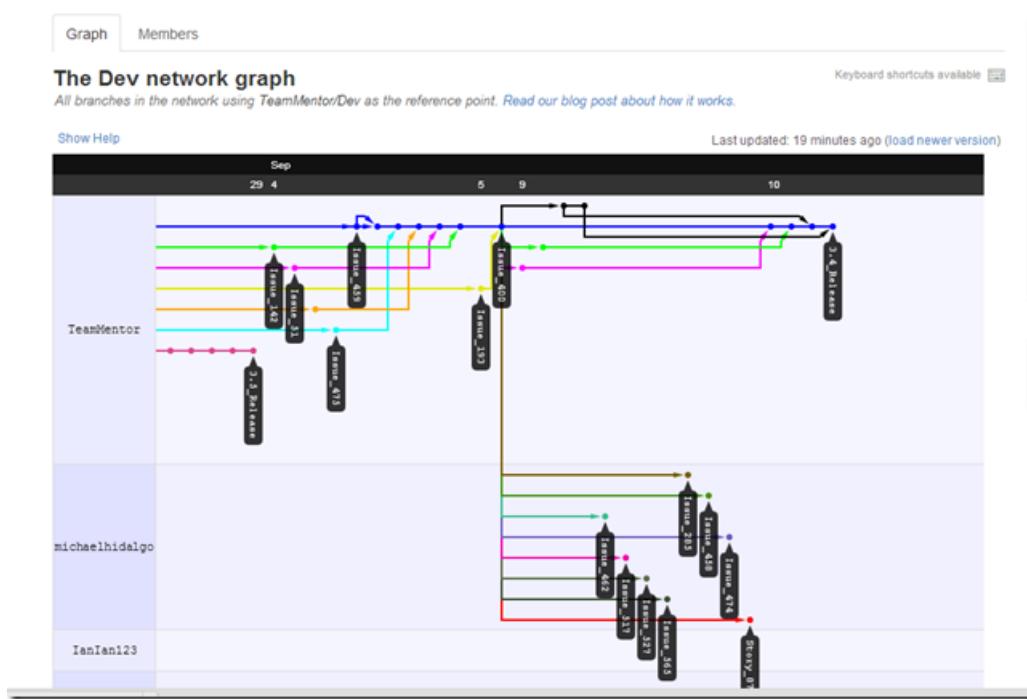


The screenshot shows a GitHub repository named "TeamMentor / Dev". The "All Requests" tab is selected, showing 21 open pull requests. The pull requests are listed in descending order of creation (Newest). Each pull request includes the title, a brief description, the user who created it, and the time ago. A "New pull request" button is located at the top right of the list. The sidebar on the right contains various icons for repository management.

Request #	Title	Author	Created
#87	Story 87 Fixing Story_87	michaelhdalgo	12 hours ago
#86	Fixing Issue 474 Fixing Issue_474 TeamMentor/Master#474	michaelhdalgo	13 hours ago
#85	Issue 458 Fixing Issue_458 TeamMentor/Master#458	michaelhdalgo	14 hours ago
#84	Issue 285 Fixing Issue_285 TeamMentor/Master#285	michaelhdalgo	14 hours ago
#83	Fixing Issue 565 https://github.com/TeamMentor/Master/issues/565 TeamMentor/Master#565	michaelhdalgo	14 hours ago
#82	Fixing Issue_527 Fixing Issue_527 TeamMentor/Master#527	michaelhdalgo	15 hours ago
#81	Fixing Issue 517 Fixing Issue_517 TeamMentor/Master#517	michaelhdalgo	16 hours ago

... which were all created using Git Branches:

³<http://blog.diniscruz.com/2013/09/git-flow-moving-patches-from-one-commit.html>



In the image above, the top lines show the commits/branches that have already been committed, and the bottom ones the branches that still need to be committed (currently on the 'open' Pull Requests)

Git Pull Request workflow

- 1) open the Pull Request page:

A screenshot of a GitHub pull request page. The title is "Fixing Issue 462". The description says "michaelhidalgo opened this pull request 16 hours ago". The status is "Open" with "+ 1 addition" and "- 1 deletion". There is one commit from "michaelhidalgo" titled "Fixing Issue 462" made 16 hours ago. A green banner at the bottom says "This pull request can be automatically merged. You can also merge branches on the command line." with a "Merge pull request" button.

2) click on the link to the issue that is being fixed:

A screenshot of the GitHub issue page for issue #462. The title is "Fixing Issue 462". The description says "michaelhidalgo opened this pull request 16 hours ago". The status is "Open" with "+ 1 addition" and "- 1 deletion". There is one participant. A blue arrow points to the link "Fixing Issue TeamMentor/Master#462" in the description area.

3) read the issue (and its history)

roman87 opened this issue 5 months ago

Fix error message in the Tbot import script

michaelhidalgo is assigned Milestone: 3.4

Error message in the import script now says that the user xml files from prior to 3.3 are looked for in Library_DataXmlDatabase\User_Data\Users instead they are looked for in Library_DataXmlDatabase\User_Data

3 participants

DinisCruz commented 3 months ago

Michael, another easy one for you

Open 8 comments Labels P4 Type: Bug With Pull R...

- 4) back in GitHub's Pull Request, click on the *Files Changed* link to see the proposed code changes:

Open michaelhidalgo wants to merge 1 commit into TeamMentor:3.4_Release from michaelhidalgo:Issue_462

Discussion Commits Files Changed

Showing 1 changed file with 1 addition and 1 deletion.

Web Applications/TM_Website/TBot/Admin/Import Legacy Users.cshtml

```

@@ -181,7 +181,7 @@ else
181   181     <span>
182   182     <hr/>
183   183
184 -  <b>Did not Found tmusers.xml</b> at location: @userFolder
184 +  <b>Did not Found tmusers.xml</b> at location: @System.IO.Directory.GetParent(userFolder).FullName
185   185   </span>
186   186 }
187   187

```

View file @ 998d155

Tip: You can add notes to lines in a file. Hover to the left of a line to make a note.

- 5) if I'm happy with the request, on the 'Discussion' tab, I click on the *Merge pull request* button

Open michaelhidalgo wants to merge 1 commit into TeamMentor:0.4_Release from michaelhidalgo:Issue_462 2 2 #80

Discussion Commits Files Changed

michaelhidalgo opened this pull request 16 hours ago

Fixing Issue 462

No one is assigned Edit
No milestone Edit

Fixing Issue TeamMentor/Master#462

1 participant

michaelhidalgo added a commit 16 hours ago

michaelhid... Fixing Issue 462 998d155

This pull request can be automatically merged. You can also merge branches on the command line.

Merge pull request

... followed by *Confirm merge*

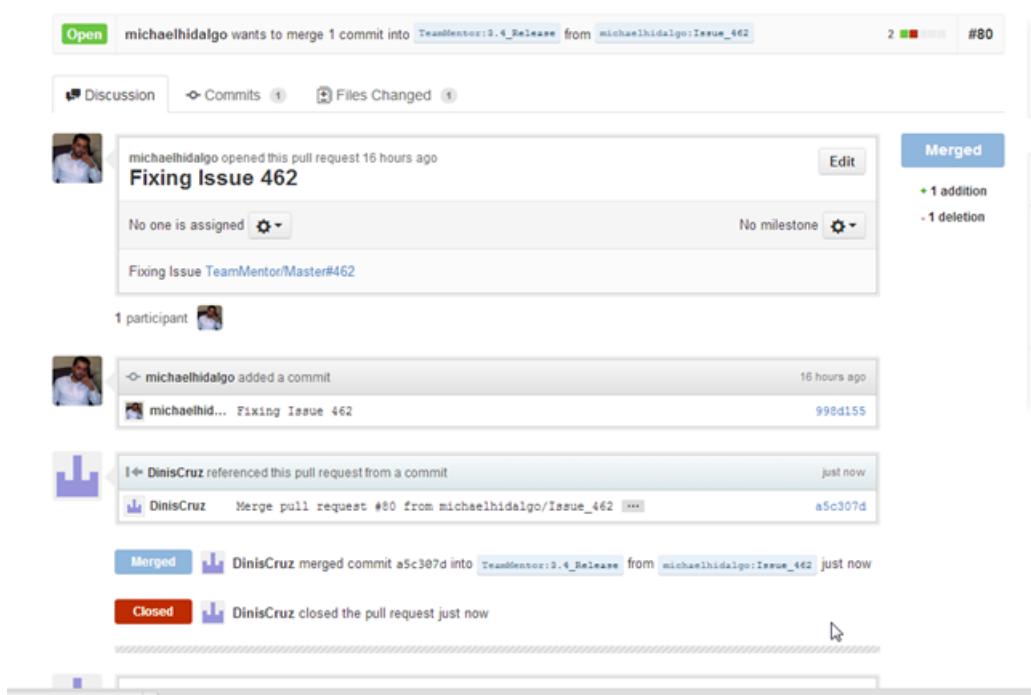
Merge pull request #80 from michaelhidalgo/issue_462

Fixing Issue 462

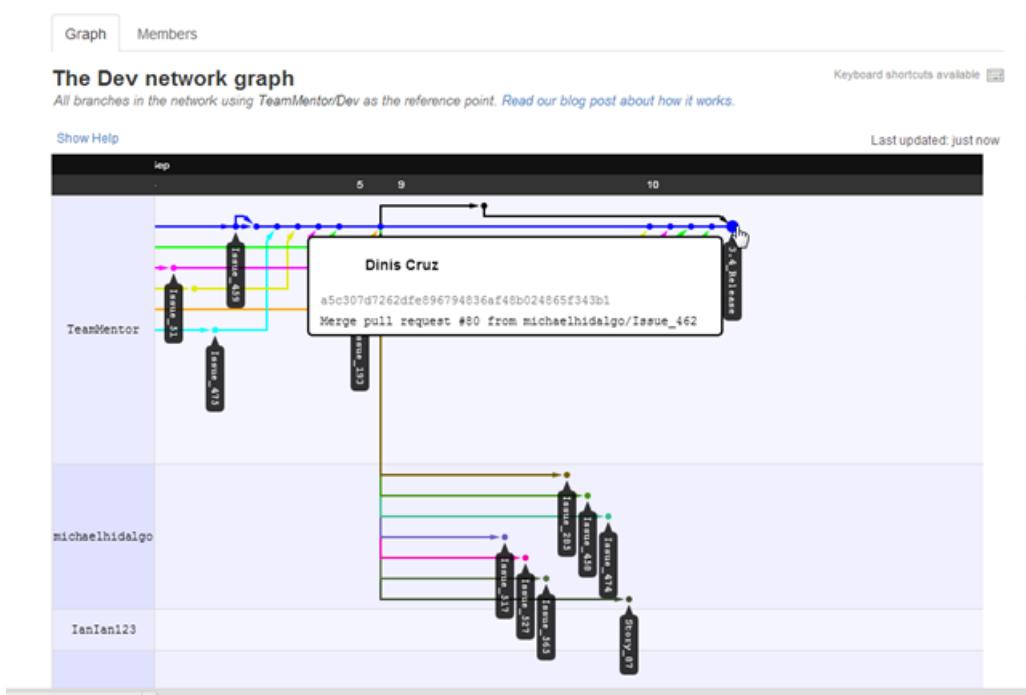
DinisCruz dinis.cruz@owasp.org

Cancel Confirm merge

- 6) optional: if this was under a repo that I owned, I would also delete the branch, in this case, Michael will have to do it on his repo/fork)



7) optional: confirm on GitHub's Network Graph that the merge happened ok (i.e. the commit is now on the **3.4_Release** branch and the **Issue_462** branch no longer is shown on Michael's fork)



- 8) optional: check that the respective issue has been correctly tagged/linked with this pull request

GitHub, Inc. [US] <https://github.com/TeamMentor/Master/issues/462>

Did not Found tmusers.xml at location: C:\TeamMentor\MichaelFork\Dev\Library_Data\XmlDatabase\User_Data

[back to commands list](#)

michaelhidalgo referenced this issue in TeamMentor/Dev a month ago

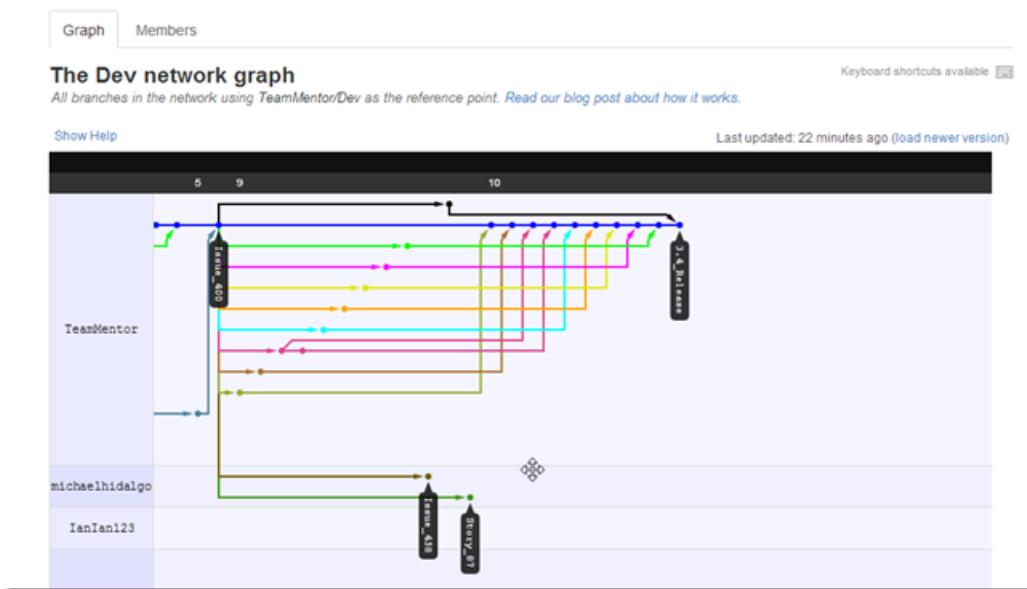
Pull Request #65: Fixing Issue_462 Closed

michaelhidalgo referenced this issue in TeamMentor/Dev 17 hours ago

Pull Request #80: Fixing Issue 462 Merged

... do this for the other Pull Requests....

Here is how the Network Graph looks like after all merges have occurred:



At the moment there are only two branches that need to be merged:

- 1) <https://github.com/TeamMentor/Dev/pull/85>⁴ : currently conflicting (i.e the merge cannot happen automatically):

⁴<https://github.com/TeamMentor/Dev/pull/85>

The screenshot shows a GitHub pull request page for issue #458. The pull request is from the branch `michaelhidalgo:Issue_458` to the `TeamMentor:3.4_Release` branch. The title of the pull request is "Fixing Issue_458 TeamMentor/Master#458". It has been opened by `michaelhidalgo` 14 hours ago. There are 8 additions and 8 deletions. The status is "Open". The pull request has 2 participants. A comment from `DinisCruz` says: "Michael, you are going to have to redo this commit, since the merge cannot happen automatically (one of your previous fixes is conflicting with it). You will need to (on your local fork): 1) pull the latest version of 3.4_Release branch 2) checkout into this Issue_458 branch 3) merge 3.4_Release into it 4) fix conflicts 5) commit the fixes 6) push Issue_458 branch. You shouldn't need to create a new Pull Request, since this page should now reflect your new changes." A note at the bottom says: "We can't automatically merge this pull request. Use the command line to resolve conflicts before continuing." A "Merge pull request" button is also visible.

- 2) <https://github.com/TeamMentor/Dev/pull/87⁵> – no idea what issue this is fixing (the link to the GitHub issue is missing)

⁵<https://github.com/TeamMentor/Dev/pull/87>

michaelhidalgo wants to merge 1 commit into `TeamMentor:3.4_Release` from `michaelhidalgo:Story_87`

Discussion [Commits](#) 1 [Files Changed](#) 1

Story 87

No one is assigned [Edit](#)

No milestone [Edit](#)

Fixing Story_87

2 participants

michaelhidalgo added a commit 13 hours ago
Fixing Story_87 814990b

DinisCruz commented 17 minutes ago

What issue is this fixing?

This pull request can be automatically merged. You can also merge branches on the command line.

Merge pull request

Hopefully this shows the power of Git and GitHub's commit/review workflow where:

- each bug has a separate Issue, Branch and Pull Request
- code review of proposed changes is really easy to do
- multiple fixes can be done in parallel with very few conflicts
- conflicts (when exist) are easy to identify and deal with
- GitHub's visualizations make a massive difference in making this workflow really smooth
- everything done by GitHub is based on git commands, which means that all actions could have been done locally, on git clones of [TeamMentor/Dev](#)⁶ and [michaelhidalgo/Dev](#)⁷

In fact, speaking of a manual step, now that we have the 3.4_Release with (just about) the final set of commits, I'm going to merge the 3.4_Release branch with the master branch (which will eventually become the release one)

⁶<https://github.com/TeamMentor/Dev>

⁷<https://github.com/michaelhidalgo/Dev>

To do that, I opened a (local) clone of TeamMentor/Dev:

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git remote -v
origin  git@github.com:TeamMentor/Dev.git (fetch)
origin  git@github.com:TeamMentor/Dev.git (push)
```

... updated it (since it is out of date with the changes made directly on the GitHub's version), using the command: **\$ git pull origin**

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git pull origin
remote: Counting objects: 2578, done.
remote: Compressing objects: 100% (1242/1242), done.
remote: Total 2303 (delta 1659), reused 1619 (delta 992)
Receiving objects: 100% (2303/2303), 11.16 MiB | 463.00 KiB/s, done.
Resolving deltas: 100% (1659/1659), completed with 228 local objects.
From github.com:TeamMentor/Dev
  d0085dc..b97a470  master      -> origin/master
 * [new branch]      3.3.3_Release -> origin/3.3.3_Release
 * [new branch]      3.4_Release -> origin/3.4_Release
 * [new branch]      3.5_Release -> origin/3.5_Release
 * [new branch]      Issue_142  -> origin/Issue_142
 * [new branch]      Issue_400  -> origin/Issue_400
 * [new tag]         v3.3.3     -> v3.3.3
From github.com:TeamMentor/Dev
 * [new tag]         v3.3.1     -> v3.3.1
Updating d0085dc..b97a470
Checking out files: 100% (38/38), done.
Fast-forward
 .gitignore
 .../ControlPanel/ManageUsers/EditUser.html
 .../TM_Website/Javascript/TM_Settings.js
 .../TM_Website/Javascript/TM_WS_Methods.js
 .../TM_Website/Javascript/gAnalytics/ga.js
 .../TM_Website/TBot/Views/User_View.cshtml
 .../bin/SecurityInnovation.TeamMentor.Website.dll
 .../TM_Website/bin/TeamMentor.CoreLib.dll
 .../Authentication/WindowsAuthentication.cs
 .../FileUpload/LibraryDownload.ashx.cs
 .../HttpHandlers/UrlRewrite/HandleUrlRequest.cs
 .../TM_AppCode/Schemas/Legacy/guidanceexplorer.cs
 .../TM_AppCode/Schemas/TMUser.cs
```

6	+ -
2	+ -
2	+ -
3	+ -
35	+ --
3	+ -
Bin	4608 -> 4608 bytes
Bin	342137 -> 346756 bytes
2	+ -
6	+ -
44	++ --
22	+ -
31	+ --

... see all branches available, using the command: **\$ git branch -a**

```
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git branch -a
 398
* master
  remotes/origin/3.3.3_Release
  remotes/origin/3.4_Release
  remotes/origin/3.5_Release
  remotes/origin/HotFix_3_3_1
  remotes/origin/Issue_142
  remotes/origin/Issue_400
  remotes/origin/NodeJS
  remotes/origin/master
```

... merged *3.4_Release* branch into *master* branch, using the command \$ git merge remotes/origin/3.4_Release

```
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git merge remotes/origin/3.4_Release
Updating b97a470..9546ef2
Fast-forward
  ./ControlPanel/Views/Admin_Links_LeftMenu.html      3 +--
  ./ControlPanel/WebServices/TM_WebServices.html      9 -----
  .../Html_Pages/Gui/Dialogs/Change_Password.html    22 ++++++-----+
  .../TM_Website/Html_Pages/Gui/Dialogs/Login.Html   12 ++++++-----
  .../Html_Pages/Gui/Panels/Right_GuidanceItem.html 3 +++
  .../Html_Pages/Gui/Panels/TopRight_Links.html     7 +++++-
  .../GuidanceItemViewer/GuidanceItemViewer.html    2 ++
  .../TM_ControlPanel/TM_ControlPanel_Open_VIEWS.js  7 +-----
  .../TM_ControlPanel/TM_ControlPanel.WebServices.js 27 -----
  .../Javascript/TM_ControlPanel/TM_ControlPanel.js  1 -
  .../Javascript/TM_Gui/TM_Gui.DataTable.js          20 ++++++-----
  .../Javascript/TM_Gui/TM_Gui.DataTableViewer.js   4 +--+
  .../TM_Website/Javascript/TM_Gui/TM_Gui.Main.js  2 +-
  .../jQuery.DataTable/dataTables.ExtraMethods.js   8 +--+-
  .../Javascript/jquery.jsTree/jquery.jstree.js     2 ++
  .../TBot/Admin/Import_Legacy_Users.cshtml         2 +-
  .../TM_Website/TBot/Views/User_Edit.cshtml       4 +++
  Web Applications/TM_Website/TM_Website.csproj     4 +---+
  .../TM_AppCode/Schemas/Models/TM_User.cs          14 ++++++-----
  .../TM_Xml_Database.Library.GuidanceItem.cs       2 +-
20 files changed, 69 insertions(+), 86 deletions(-)
delete mode 100644 Web Applications/TM_Website/Html_Pages/ControlPanel/WebServi
delete mode 100644 Web Applications/TM_Website/Javascript/TM_ControlPanel/TM.Co
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
```

... pushed these changes into GitHub, using the command \$ git push origin master:master

```
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git push origin master:master
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
  b97a470..9546ef2  master -> master
```

(note how no files were changed with this push, since all data was already in the *3.4_Release*

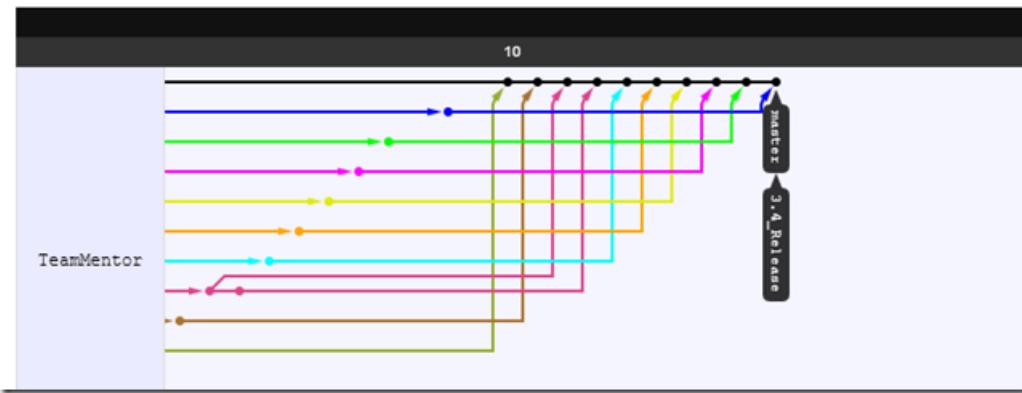
branch, this commit was just saying to GitHub's version: '*please point the master branch into the 3.4_Release commit*'

After this commit, GitHub's network graph will show that the **master** branch is now at the same commit as the **3.4_Release** branch

The Dev network graph

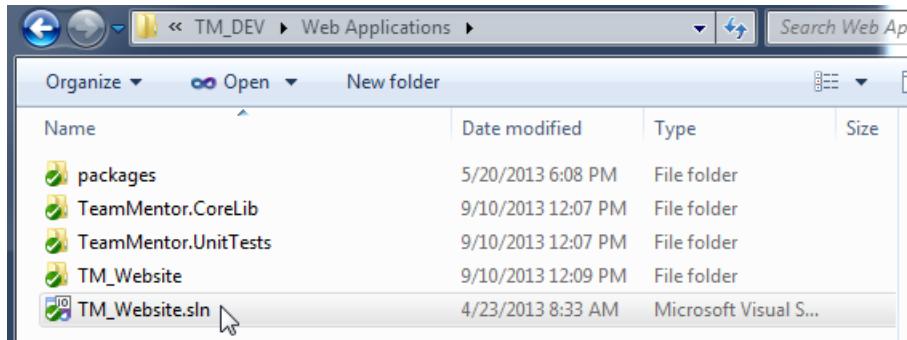
All branches in the network using TeamMentor/Dev as the reference point. Read our blog post about how it works.

Show Help

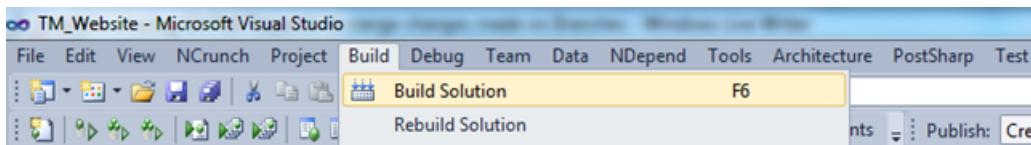


But we are not done here, we will still need to update the compiled TeamMentor DLLs (and see if any UnitTests broke)

Let's start by opening up the solution file in VisualStudio 2010:



... then clean and build the solution:



... which succeeded ok:

```

Output
Show output from: Build
-----
----- Build started: Project: TeamMentor.CoreLib, Configuration: Debug Any CPU -----
: message : PostSharp complete -- 0 errors, 0 warnings, served in 4464 ms
TeamMentor.CoreLib -> E:\TeamMentor\TM_Dev Repos\TM_DEV\Web Applications\TeamMentor.CoreLib\b...
----- Build started: Project: TeamMentor.UnitTests, Configuration: Debug Any CPU -----
: message : PostSharp complete -- 0 errors, 0 warnings, served in 1944 ms
TeamMentor.UnitTests -> E:\TeamMentor\TM_Dev Repos\TM_DEV\Web Applications\TeamMentor.UnitTest...
----- Build started: Project: TM_Website, Configuration: Debug Any CPU -----
TM_Website -> E:\TeamMentor\TM_Dev Repos\TM_DEV\Web Applications\TM_Website\bin\SecurityInno...
===== Build: 3 succeeded or up-to-date, 0 failed, 0 skipped =====

```

Next, change the version number to *TM 3.4 – Dev 20*

```

Settings.js* X
//TM Settings
window.TM.tmVersion      = "TM 3.4 - Dev 20";
window.TM.ArticleTitle    = "TeamMentor 3.3";
|
```

And start TM locally (just to see is all looks good):

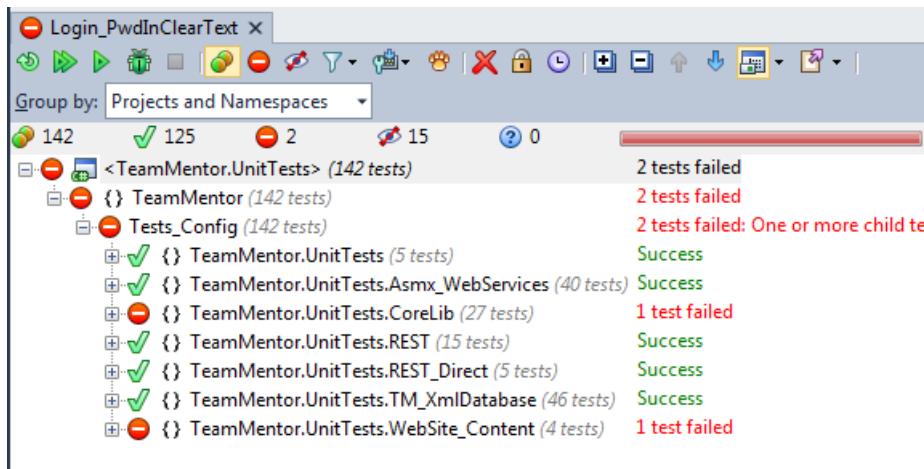
The screenshot shows the TeamMentor web interface. On the left, there's a sidebar with 'Applied Filters' showing 'OWASP' selected under 'Guidance Libraries'. The main area displays a search result for 'A Secure Key Storage Location Is Used'. The results table has columns: Title, Technology, Phase, Type, and Category. One row is highlighted in blue. To the right, there's a 'Selected Article' panel showing the details of the selected article.

Title	Technology	Phase	Type	Category
A Secure Key Storage Location Is Used	ASP.NET 3.5	Implementation	Checklist Item	Encryption
Access to the Credential Store is Limited	Java	Implementation	Checklist Item	Authentication
Account Administration Functionality Is Secured	Java	Design	Checklist Item	Authorization
Accounts Are Locked After Consecutive Failed Login Attempts	Java	Design	Checklist Item	Authentication
Administrative Interfaces Are Enumerated And Secured	Any	Deployment	Checklist Item	Server Hardening

Now, its time to run all UnitTests (in this case using ReSharper NUnit plugin):



... with two tests failing:



The first one was easy to fix (it was a case of updating the UnitTests to the changes made to the *TM_User* required fields):

```
[Test]
public void Validation_TM_User()
{
    var tmUser = new TM_User();
    //var requiredValues = "Company,Country,FirstName,LastName,State,Title,UserName,Email".split(",");
    var requiredValues = "UserName,Email".split(",");
    var validationResults = tmUser.validate();
    var resultsMapped = validationResults.indexed_By_MemberName();
    var validationok = tmUser.validation_Ok();
```

The 2nd one was caused because the Google Analysis file has changed:



Here is the test that does this check:

```
[Test]
public void Check_GoogleAnalytics()
{
    var assembly      = this.type().Assembly;

    var dllLocation   = assembly.CodeBase.subString(8);
    var webApplications = dllLocation.parentFolder()
                           .pathCombine(@"..\..\..\..");
    var tmWebsite     = webApplications.pathCombine("TM_Website");
    var gAnalyticsFolder = tmWebsite.pathCombine(@"Javascript\gAnalytics");
    var gaFile        = gAnalyticsFolder.pathCombine("ga.js");

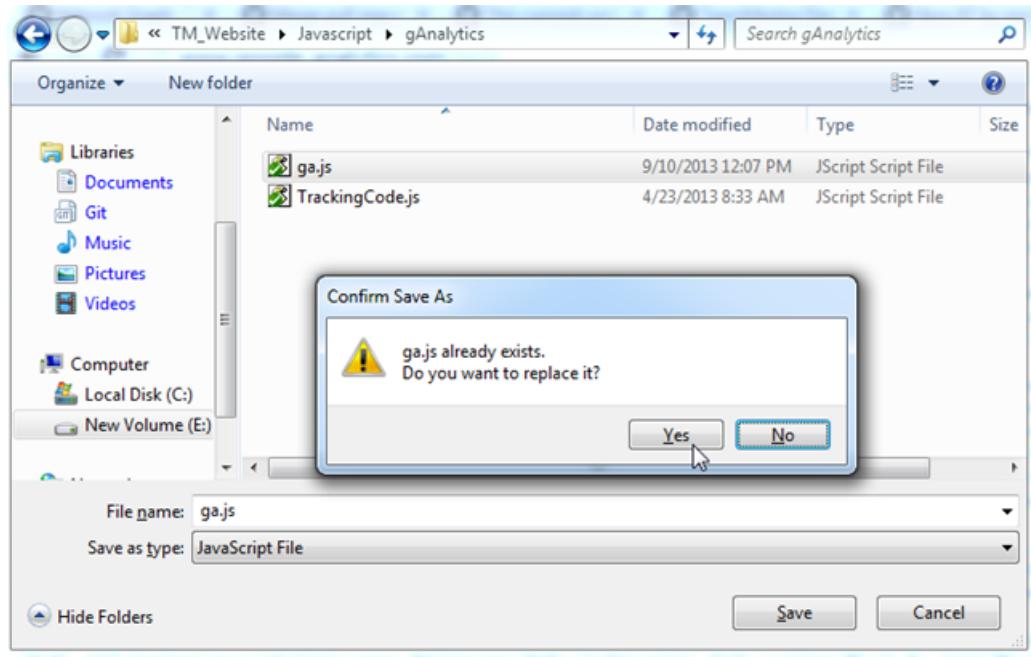
    Assert.That(dllLocation   .fileExists(), "dllLocation file");
    Assert.That(webApplications.dirExists(), "webApplications dir");
    Assert.That(tmWebsite     .dirExists(), "tmWebsite dir");
    Assert.That(gAnalyticsFolder.dirExists(), "gAnalyticsFolder dir");
    Assert.That(gaFile        .fileExists(), "gaFile file");

    var tmVersion      = gaFile.fileContents().fixCRLF();

    Assert.That(tmVersion.valid()  , "ga.js tmVersion not valid");
    var googleVersion = "http://www.google-analytics.com/ga.js".GET().fixCRLF();
    Assert.That(googleVersion.valid(), "ga.js googleVersion not valid");
    Assert.AreEqual(tmVersion, googleVersion, "ga.js files didn't match");
}
```

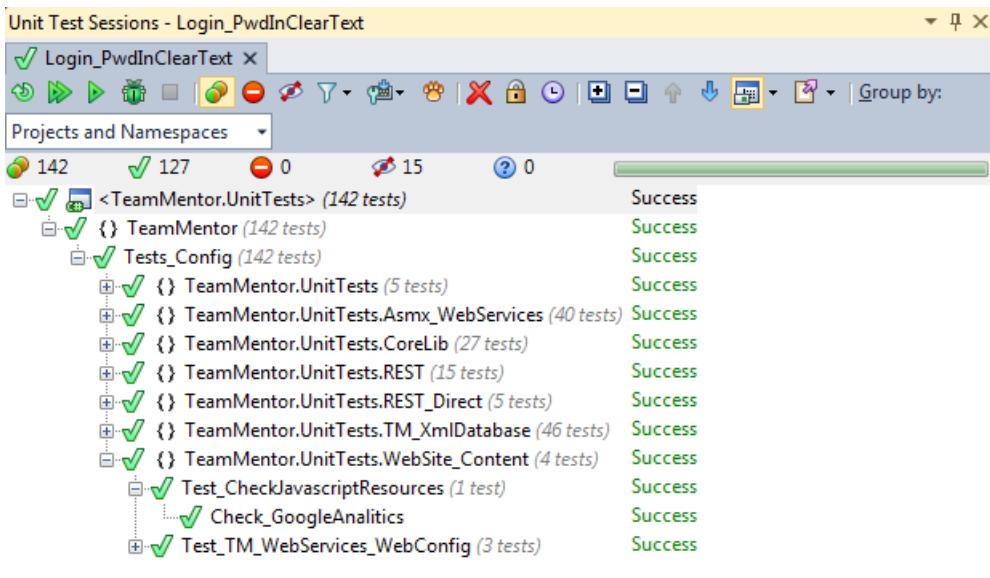
... which basically checks that the <http://www.google-analytics.com/ga.js>⁸ we are using is still the same one served by google (this is a good security practice since TeamMentor's security is not dependent on Google's server).

The fix is to update that file:



... and rerun all tests (just to confirm it):

⁸<http://www.google-analytics.com/ga.js>



Committing changes made locally.

Keeping up with the model of only doing commits on branches, I quickly created a new branch, using the command: `$ git checkout -b 3.4_Dll_Updates`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git checkout -b 3.4_Dll_Updates
M   Web Applications/TM_Website/Javascript/TM/Settings.js
M   Web Applications/TM_Website/Javascript/gAnalytics/ga.js
M   Web Applications/TM_Website/bin/SecurityInnovation.TeamMentor.Website.dll
M   Web Applications/TM_Website/bin/TeamMentor.CoreLib.dll
M   Web Applications/TeamMentor.UnitTests/CoreLib/Test_Schemas.cs
Switched to a new branch '3.4_Dll_Updates'
```

(note how the small changes I made were also marked as ‘Modified’ (namely the version change, the UnitTests fixes and the recompiled dlls)

... added the files to be committed using the command: `$ git add .`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (3.4_Dll_Updates)
$ git add .
warning: LF will be replaced by CRLF in Web Applications/TM_Website/Javascript/gAnalytics/ga.js.
The file will have its original Line endings in your working directory.
```

... created an commit using the command: `$ git commit -m 'Changing version, adding compiled Dlls, fixing couple UnitTests'`

```
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (3.4_Dll_Updates)
$ git commit -m 'Changing version, adding compiled DLLs, fixing couple UnitTests'
[3.4.Dll_Updates warning: LF will be replaced by CRLF in Web Applications/TM_Website/Javascript/
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in Web Applications/TM_Website/Javascript/gAnalytics/ga.js.
The file will have its original line endings in your working directory.
8300522] Changing version, adding compiled DLLs, fixing couple UnitTests
warning: LF will be replaced by CRLF in Web Applications/TM_Website/Javascript/gAnalytics/ga.js.
The file will have its original line endings in your working directory.
 5 files changed, 67 insertions(+), 67 deletions(-)
 rewrite Web Applications/TM_Website/Javascript/gAnalytics/ga.js (80%)
 rewrite Web Applications/TM_Website/bin/TeamMentor.CoreLib.dll (92%)
```

... pushed this branch to GitHub (not 100% necessary, but it will help with the graph), using the command **\$ git push origin 3.4_Dll_Updates:3.4_Dll_Updates**

... applied these changes to the 3.4_Release branch (locally and at GitHub), using the commands:

- **\$ git push origin 3.4_Dll_Updates:3.4_Dll_Updates**
- **\$ git checkout 3.4_Release**
- **\$ git merge 3.4_Dll_Updates**
- **\$ git push origin 3.4_Release:3.4_Release**

```
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (3.4_Dll_Updates)
$ git push origin 3.4_Dll_Updates:3.4_Dll_Updates
Counting objects: 29, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 125.18 KiB | 0 bytes/s, done.
Total 15 (delta 12), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 * [new branch]      3.4_Dll_Updates -> 3.4_Dll_Updates

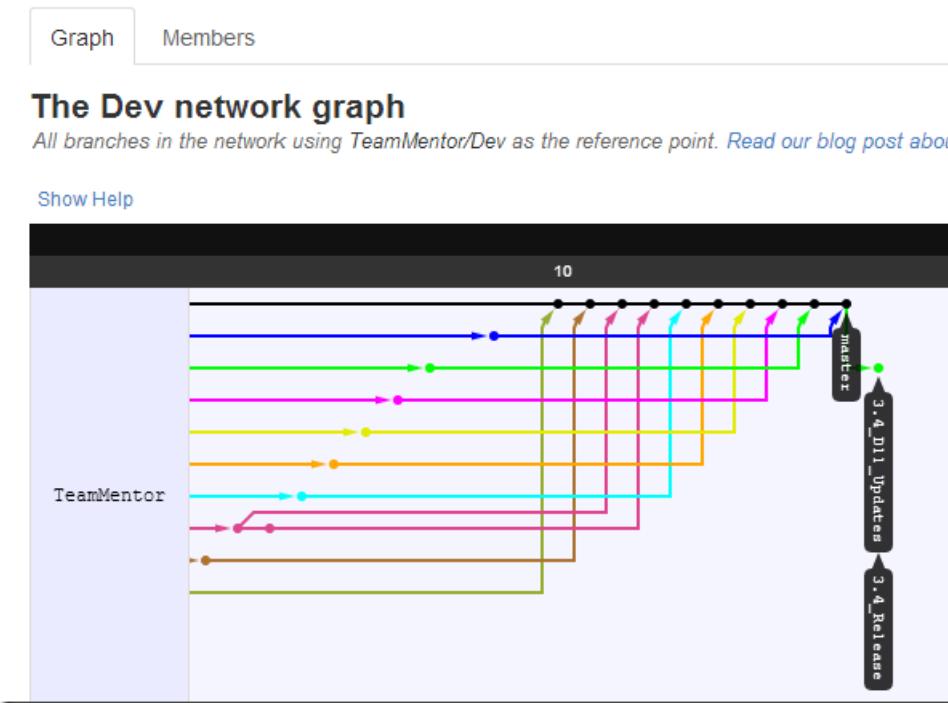
b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (3.4_Dll_Updates)
$ git checkout 3.4_Release
Branch 3.4_Release set up to track remote branch 3.4_Release from origin.
Switched to a new branch '3.4_Release'

b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (3.4_Release)
$ git merge 3.4_Dll_Updates
Updating 9546ef2..8300522
Fast-forward
 .../TM_Website/Javascript/TM_Settings.js           |    2 +-+
 .../TM_Website/Javascript/gAnalytics/ga.js          | 115 ++++++-----+
 .../bin/SecurityInnovation.TeamMentor.Website.dll | Bin 4608 -> 4608 bytes
 .../TM_Website/bin/TeamMentor.CoreLib.dll          | Bin 346756 -> 346751 bytes
 .../TeamMentor.UnitTests/CoreLib/Test_Schemas.cs   |    3 +-+
 5 files changed, 60 insertions(+), 60 deletions(-)

b2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (3.4_Release)
$ git push origin 3.4_Release:3.4_Release
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
 9546ef2..8300522  3.4_Release -> 3.4_Release
```

(note that this is an example of a ‘manual Pull Request’)

A quick look at GitHub’s network graph, shows the **3.4_Release** branch at the same commit as **3.4_Dll_Updates** branch (both one commit behind the **master** branch)



Finally we update master with these changes, using the commands:

- `$ git checkout master`
- `$ git merge 3.4_Release`
- `$ git push origin master:master`

```
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (3.4_Release)
$ git checkout master
Switched to branch 'master'

o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git merge 3.4_Release
Updating 9546ef2..8300522
Fast-forward
.../TM_Website/Javascript/TM/Settings.js      2 +-+
.../TM_Website/Javascript/gAnalytics/ga.js     115 ++++++-
.../bin/SecurityInnovation.TeaMmentor.Website.dll Bin 4608 -> 4608 bytes
.../TM_Website/bin/TeamMentor.CoreLib.dll       Bin 346756 -> 346751 bytes
.../TeamMentor.UnitTests/CoreLib/Test_Schemas.cs 3 +-
5 files changed, 60 insertions(+), 60 deletions(-)

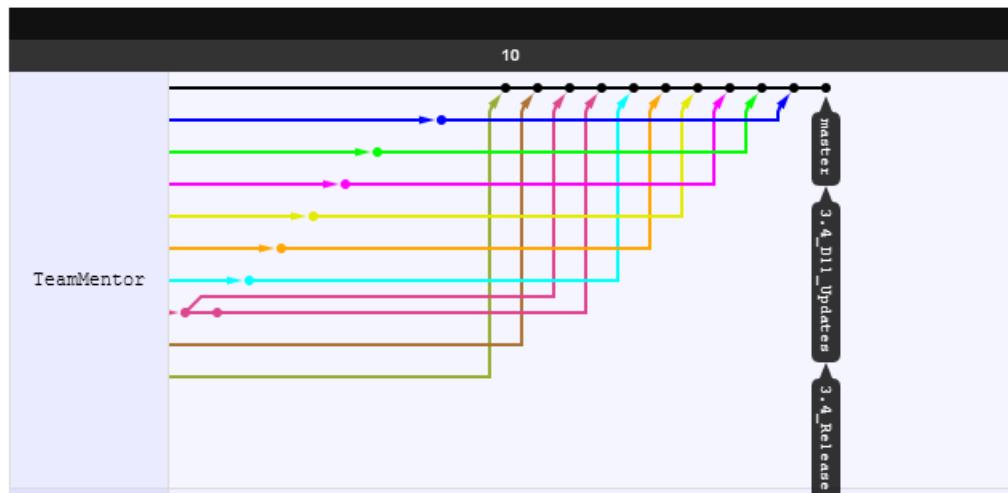
o2@WIN-FGNQ5AARJ80 /e/TeamMentor/TM_Dev Repos/TM_DEV (master)
$ git push origin master:master
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
  9546ef2..8300522  master -> master
```

... and now all branches are at the same level:

The Dev network graph

All branches in the network using TeamMentor/Dev as the reference point. [Read our blog post about how it](#)

[Show Help](#)



Testing QA version created by TeamCity and deployed to Azure:

As shown in [past blog posts](#)⁹, we also have TeamCity configured to monitor TeamMentor commits and auto-publish new builds into Azure.

⁹<http://blog.diniscruz.com/search/label/TeamCity>

In this case after the latest commit into GitHub TeamMentor/Dev master repo, TeamCity picked up the changes and:

- * Built the code
- * Published to Azure
- * Run all unit-tests

At the moment there is one unit failing:

The screenshot shows a TeamCity build interface. At the top, it says "TeamMentor > TeamMentor - Master 3.4" and "Build #358 (10 Sep 13 12:15)". Below that is a navigation bar with tabs: Overview, Changes (selected), Tests, Build Log, Build Parameters, and Artifacts. On the right, there are buttons for "Run", "Build Actions", "Edit Configuration Settings", and links for "#357", "All history", and "Last recorded build". The main content area has a red border around the failure details. It shows the following information:

- Result: 1 Tests failed: 1, passed: 126, ignored: 15
- Time: 10 Sep 13 12:15 - 12:22 (6m.29s)
- Investigation: Start investigation... of current problems in this build configuration (TeamMentor - Master 3.4)

Below this, a grey bar indicates "1 test failed (no new)". Under "Tests", it lists "TeamMentor.UnitTesting.dll: TeamMentor.UnitTesting.TM_XmlDatabase (1)" and "Test_Library_SaveToDisk.TestUseOfTempFolders".

... which doesn't look problematic (it feels like a TeamCity specific case).

The Azure deployment went ok:

The screenshot shows the Azure Deployment Center. At the top, it has a navigation bar with "DASHBOARD", "DEPLOYMENTS" (selected), "MONITOR", "CONFIGURE", "SCALE PREVIEW", and "LINKED RESOURCES". Below that is a section titled "deployment history". It shows a deployment log entry:

- ACTIVE DEPLOYMENT: Tuesday, September 10, 2013 1:17 PM
- Changing version, adding compiled DLLs, fixing couple UnitTests
- ID: 83005224b2 AUTHOR: Dinis Cruz DEPLOYED BY: tmci

A red icon with a white diamond shape is on the left.

With a clean version of TM ready for testing:

The screenshot shows the TeamMentor web interface. On the left, there's a sidebar titled "Applied Filters" and "Guidance Libraries". The main area has a search bar at the top. Below it are four dropdown filters: Technology (Java), Phase (Design, Implementation, Test), Type (Checklist Item, Code Example, Guideline, How To), and Category (Data Sanitization, Error Handling, Input and Data, Session Management). A message below the filters says "Showing 0 items (out of 0)" and "No data available in table". At the bottom of the main area, there are sorting columns for Title, Technology, Phase, Type, and Category. On the right, there's a "Selected Article" panel with a "Login required" message and a form for entering login credentials: Username, Password, Repeat Password, E-Mail, First Name, Last Name, and Company.

...easily populated with a couple libraries:

This screenshot shows the same TeamMentor interface after adding some data. In the "Guidance Libraries" sidebar, under the Java section, there are several items listed: Fundamentals of Security, OWASP Top 10 2010, PCI DSS Code Review, PCI DSS Compliance, Security Engineering, Top 5 Rich Client Vulnerabilities, and Top 5 Web Service Vulnerabilities. The search results table now shows 63 items. The first few entries are: "A Global Exception Handler Is Used for Unhandled Exceptions" (Java, Implementation, Checklist Item, Error Handling); "Add Unique Tokens to HTTP Requests Using ESAPI" (Java, Implementation, Code Example, Session Management); "Add Unique Tokens to HTTP Requests Using Java Server Faces (JSF)" (Java, Implementation, Code Example, Session Management); and "Add Unique Tokens to HTTP Requests Using Plain Java Server Pages (JSP)" (Java, Implementation, Code Example, Session Management). The "Selected Article" panel on the right displays the details for the first article, including its title, applies to Java web-based applications using JSF, summary about preventing CSRF, and objectives related to CSRF attacks.

Running TM locally from Zip File

A final test is to go to the main [https://github.com/TeamMentor/Dev¹⁰](https://github.com/TeamMentor/Dev) site and click on the [Download Zip¹¹](#) button:

¹⁰<https://github.com/TeamMentor/Dev>

¹¹<https://github.com/TeamMentor/Dev/archive/master.zip>

The screenshot shows the GitHub repository page for 'TeamMentor / Dev'. At the top, it displays '695 commits', '8 branches', '7 releases', and '4 contributors'. Below this, a list of recently pushed branches is shown: '3.4_DLl_Updates' (9 minutes ago) and '3.4_Release' (9 minutes ago), each with a 'Compare & pull request' button. The main commit list shows the following entries:

Author	Commit Message	Date
DinisCruz	authoried 9 minutes ago	latest commit 83006224b2
Docs	adding v3.1 of the ReleaseNotes.v3.1.txt	11 months ago
Tools	adding pinned functionality to Applied filters	a year ago
Web Applications	Changing version, adding compiled Dlls, fixing couple UnitTests	9 minutes ago
.gitignore	Refactored method to remove sessions and added method reset all sessions	3 months ago
README.md	added ReleaseNotes for v3.2 and moved to /Docs folder	11 months ago
start TeamMentor.bat	Set version to 3.2	11 months ago

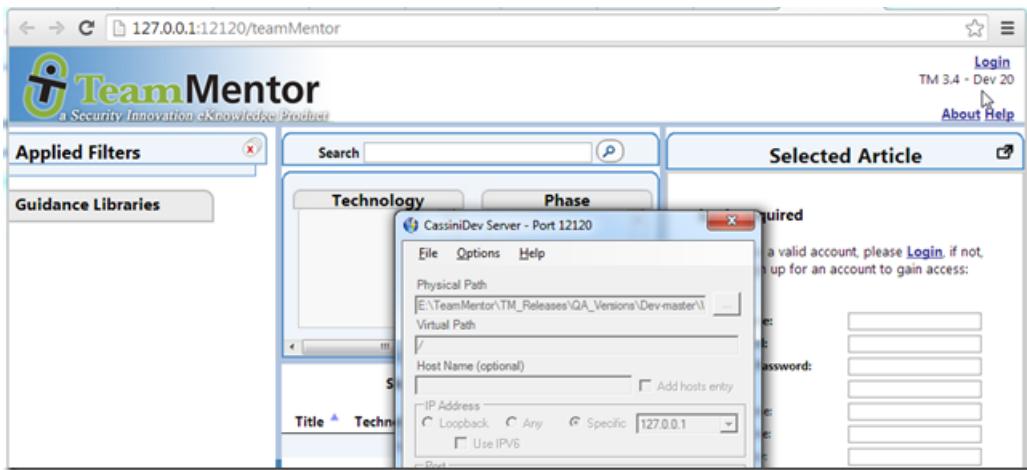
At the bottom of the commit list is a link to 'README.md'. On the right side of the page, there's a sidebar with links for 'Code', 'Issues', 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', 'Network', and 'Settings'. Below the sidebar are download options: 'HTTPS clone URL' (https://github.com/TeamMentor/Dev), 'Clone in Desktop', and 'Download ZIP'.

...extract the zip files into a local folder, and click on the 'start TeamMentor.bat' file:

The screenshot shows a Windows File Explorer window displaying the contents of a folder path: 'New Volume (E:) \ TeamMentor \ TM_Releases \ QA_Versions \ Dev-master'. The folder structure is as follows:

Name	Date modified	Type	Size
Docs	9/10/2013 12:53 PM	File folder	
Tools	9/10/2013 12:53 PM	File folder	
Web Applications	9/10/2013 12:53 PM	File folder	
.gitignore	9/10/2013 12:39 PM	Text Document	1 KB
README.md	9/10/2013 12:39 PM	MD File	1 KB
start TeamMentor.bat	9/10/2013 12:39 PM	Windows Batch File	1 KB

...which will start the .Net Cassini webserver on port 12120:



(note how the version number matches the commit made earlier)

At this stage:

- TM is just about ready for a final round of QA
- TM 3.4 RC1 will be created as soon as:
 - Michael fixes the couple pending issues
 - All UI UnitTests pass
 - All backend UnitTests pass (shown above)

Script to Git Clone 13 repositories in order to have all TeamMentor Libraries in one folder

Part of the push for the 3.4 release of [TeamMentor](#)¹², I wanted to have a copy of all TeamMentor libraries locally (there are 13 libraries on the 3.4 release).

Since O2 Platform's FluentSharp has native Git support, I was able to do create the clones using this script (note how simple it is to create a clone from a GitHub repo):

```
1 var baseFolder = @"E:\TeamMentor\Libraries\SI Library";
2 var contentRepo = "git@github.com:TMContent/{0}.git";
3 var libraries = new [] { "Lib_PHP", "Lib_CWE", "Lib_iOS", "Lib_Android", "Lib_PCI_DSS_Compliant\
4 ce",
5                         "Lib_.NET_4.0", "Lib_.NET_3.5", "Lib_.NET_2.0", "Lib_Java", "Lib_CPP",
6                         "Lib_Vulnerabilities", "Lib_Scala", "Lib_HTML5" };
7
8 var stopWatch = utils.new_Stopwatch();
9 foreach(var library in libraries)
10 {
11     var gitRepo      = contentRepo.format(library);
12     var targetFolder = baseFolder.pathCombine(library);
13     if (targetFolder.isNotGitRepository())
14         gitRepo.git_Clone(targetFolder);
15 }
16 return "Cloning took: " + stopWatch.stop().minutes_Seconds_And_Miliseconds();
17
18 //using FluentSharp.Git
19 //O2Ref:FluentSharp.NGit.dll
```

The script takes about 1m to run:

¹²<https://teammentor.net/>

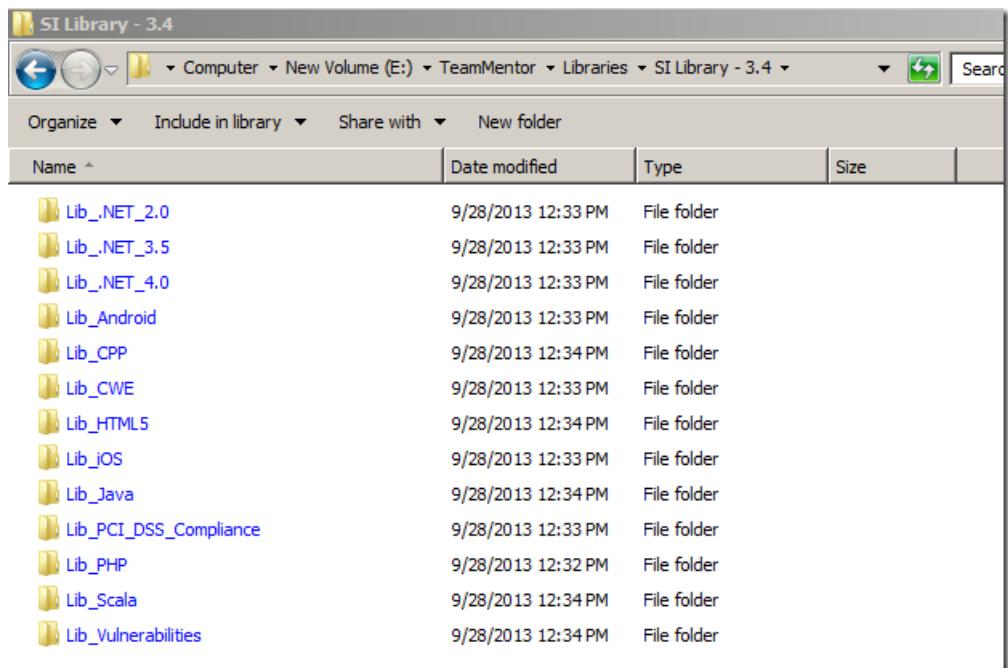
The screenshot shows the 'Inspector' application interface. On the left, under 'Command To Execute', is a block of JavaScript code that clones several GitHub repositories into a local folder. On the right, under 'Invoke and Result', there is a log viewer window showing the execution results.

```

Command To Execute
1 var baseFolder = @"E:\TeamMentor\Libraries\SI Library - 3.4";
2 var contentRepo = "git@github.com:TMContent/{0}.git";
3 var libraries = new [] { "Lib_PHP", "Lib_CWE", "Lib_iOS", "Lib_Android", "Lib_PCI_DSS_Compliance",
4                         "Lib_.NET_4.0", "Lib_.NET_3.5", "Lib_.NET_2.0", "Lib_Java", "Lib_CPP",
5                         "Lib_Vulnerabilities", "Lib_Scala", "Lib_HTML5" };
6
7 var stopWatch = utils.new_Stopwatch();
8 foreach(var library in libraries)
9 {
10     var gitRepo = contentRepo.format(library);
11     var targetFolder = baseFolder.pathCombine(library);
12     if (!targetFolder.isNotGitRepository())
13         gitRepo.git_Clone(targetFolder);
14 }
15 return "Cloning took: " + stopWatch.stop().minutes_Seconds_And_Miliseconds();
16
17 //using FluentSharp.Git
18 //O2Ref:FluentSharp.NGIt.dll
19
Log Viewer
[12:34:29 PM] DEBUG: [API_NGIt] clone completed in: 00:00:07.5900000
[12:34:28 PM] INFO: [GitProgress] Begin Task: Updating references : 1
[12:34:28 PM] INFO: [GitProgress] Begin Task: Resolving deltas : 2588
[12:34:28 PM] INFO: [GitProgress] Begin Task: Remote: https://github.com/TeamMentor/.../0.03
[12:34:24 PM] INFO: [GitProgress] Begin Task: remote Compressing objects : 2338
[12:34:24 PM] INFO: [GitProgress] Begin Task: remote Counting objects : 0
[12:34:21 PM] DEBUG: [API_NGIt] cloning: git@github.com:TMContent/Lib_HTML5.git into E:\TeamMentor\Libraries\SI Library - 3.4\Lib_HTML5
[12:34:21 PM] DEBUG: [API_NGIt] clone completed in: 00:00:09.5980000
[12:34:21 PM] INFO: [GitProgress] Begin Task: Updating references : 1

```

And the end result is a folder with all libraries cloned:



With each folder containing the git repository for that library

Name	Date modified	Type	Size
.git	9/28/2013 12:33 PM	File folder	
Attack	9/28/2013 12:33 PM	File folder	
Checklist Item	9/28/2013 12:33 PM	File folder	
Code Example	9/28/2013 12:33 PM	File folder	
Guideline	9/28/2013 12:33 PM	File folder	
How To	9/28/2013 12:33 PM	File folder	
Inspection Question	9/28/2013 12:33 PM	File folder	
Principle	9/28/2013 12:33 PM	File folder	
Question and Answer	9/28/2013 12:33 PM	File folder	
.NET 2.0.xml	9/28/2013 12:33 PM	XML Document	341 KB
README.md	9/28/2013 12:33 PM	MD File	1 KB

Next, I zipped all these files into the **SI Library -3.4.zip** file (note that they all must be on the root of the zip)

Name	Size	Packed Size
Lib_.NET_2.0	10 068 355	5 584 436
Lib_.NET_3.5	11 229 681	5 906 184
Lib_.NET_4.0	4 762 383	2 434 917
Lib_Android	296 820	166 970
Lib_CPP	3 276 161	1 738 774
Lib_CWE	592 679	327 451
Lib_HTML5	2 054 968	1 403 439
Lib_iOS	290 236	146 143
Lib_Java	5 591 563	2 834 280
Lib_PCI_DSS_Compliance	1 286 349	740 844
Lib_PHP	4 467 277	2 314 023
Lib_Scala	4 713 881	2 892 101
Lib_Vulnerabilities	1 607 389	1 184 956

Then, on a local QA TM instance, I:

- went into the admin panel,
- chose up upload the zip,
- triggered the installation (i.e. unzip) of those libraries
- rebuilt the cache:

Execution Parameters

WebMethod: XmlDatabase_ReloadData
Params: []

Execution Result

```
[In the library 'E:\\TeamMentor\\TM_Releases\\Master_3_4\\Library_Data\\XmlDatabase' there are 13 library(ies), 369 views and 6509 G
```

Once that was completed, a reload of the home page shows the 13 libraries:

The screenshot shows the TeamMentor application interface. At the top, there's a header bar with links for Edit Mode, Change Password, Control Panel, Logout, TM 3.4 - RC 4, and a user logged in as 'admin'. Below the header is a search bar and a navigation bar with tabs for Applied Filters, Selected Article, and a sidebar.

Applied Filters:

- Technology: Any, .NET 2.0, .NET 2.0, ADO.NET 2.0, ASP.NET 2.0, Any, SQL Server 2000
- Phase: Deployment, Design, Implementation, Test
- Type: Attack, Checklist, Code Example, Guideline, How To, Inspection, Procedure

Selected Article:

A Centralized Log Server Is Deployed

Applies to:

- PCI DSS Requirement 10.5
- PCI DSS Requirement 10.5.1
- PCI DSS Requirement 10.5.2
- PCI DSS Requirement 10.5.3
- PCI DSS Requirement 10.5.4
- PCI DSS Requirement 10.6
- PCI DSS Requirement 10.6.a
- PCI DSS Requirement 10.6.b
- PCI DSS Requirement 10.7
- PCI DSS Requirement 10.7.a
- PCI DSS Requirement 10.7.b

What to Check For:
A centralized log server should be deployed.

Why:
It may be possible to satisfy PCI DSS requirements without maintaining a centralized log server, but

Including the new *Html5* library:

The screenshot shows the TeamMentor web interface at localhost:3187/teamMentor. The left sidebar displays a tree view of 'Guidance Libraries' under 'Applied Filters', including categories like .NET, Android, C++, CWE, HTML5, iOS, Java, PCI DSS Compliance, PHP, Scala, and Vulnerabilities. The main content area features a search bar and three filter panels: Technology (HTML5, Web Application), Phase (Design, Implementation), and Type (Checklist, Guideline, Principle). Below these filters, a table lists 164 items. The first four items in the list are:

Title	Technology	Phase	Type	Category
A Control Flow Analysis Is Performed	Web Application	Implementation	Checklist Item	Security Engineering
A Dataflow Analysis Is Performed	Web Application	Implementation	Checklist Item	Security Engineering
A Filter Is Used to Inject X-Frame Options	HTML5	Implementation	Checklist Item	HTTP
A Secure Administrative Interface that Allows Managing Cryptographic Keys Is	Web Application	Implementation	Checklist Item	Cryptography

The right panel, titled 'Selected Article', discusses 'A Control Flow Analysis Is Performed'. It includes sections for 'What to Check For', 'Why', 'How to Check', and 'How to Fix'.

... and the new *Scala* library

The screenshot shows the TeamMentor application interface. At the top, there's a header with a logo, the title "TeamMentor", and a sub-header "a Security Innovation eKnowledge Product". Below the header, there's a toolbar with links for "Edit Mode", "Change Password", "Control Panel", "Logout", and "TM 3.4 - RC 4". A user is logged in as "admin".

Applied Filters: Shows filters for "Technology" (Any, Scala with Play ...) and "Phase" (Any, Deployment, Design). The search bar shows "Showing 11 items (out of 11)".

Guidance Libraries: A tree view of security guidelines categorized by technology. The "Scala" category is expanded, showing the "OWASP Top 10 - 2010" section with 10 items listed.

Title	Technology	Phase
A Strong Application Architecture Is Used	Any	Design
Design a Strong Application Architecture	Any	Design
Directory Browsing Is Disabled	Scala with Play Framework	Deployment
Disable Directory	Scala with Play	Deployment

Selected Article: Displays a step-by-step guide for configuring Nginx. Step 2 discusses finding the default configuration file and step 3 discusses enabling directory browsing.

```

server {
    listen 80;
    server_name domain.com www.domain.com;
    access_log /var/log/nginx/access.log;
    root /path/to/root;
    location / {
        index index.html index.htm;
    }
    location /somemdir {
        autoindex on;
    }
}

```

3. Ensure directory browsing is

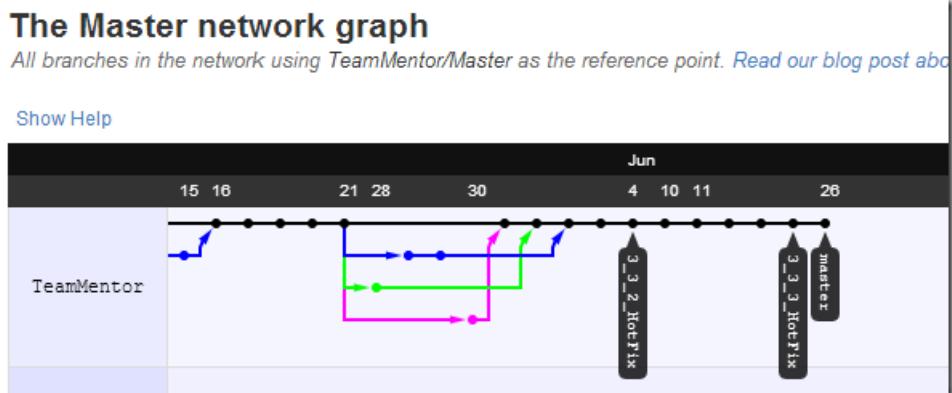
11. October 2013

- Fixing the Merge conflict caused by one extra commit on TeamMentor master
- Enabling GitHub Two Factor Authentication
- Syncing all releases to the same commit and Tag (for TeamMentor v3.4)

Fixing the Merge conflict caused by one extra commit on TeamMentor master

On the 3.4 Release of [TeamMentor¹](#) (which was the first release we really used [Git Flow²](#) on development (see this great presentation on [Git Branching Model³](#)) we ended up with a situation where the commit that was the parent of all feature/fix branches was off-by-one the master of the [TeamMentor/Master⁴](#) repository (we also had to do a bunch of back-porting of fixes into that commit, see [Git Flow - Moving patches from one Commit into another Commit post⁵](#))

In practice this means that the [TeamMentor/Master⁶](#) graph currently looks like this:



... with the master branch on the commit [fe26934d489e65660bd67be7811effcbccad1d19](#)

¹<https://teammentor.net/>

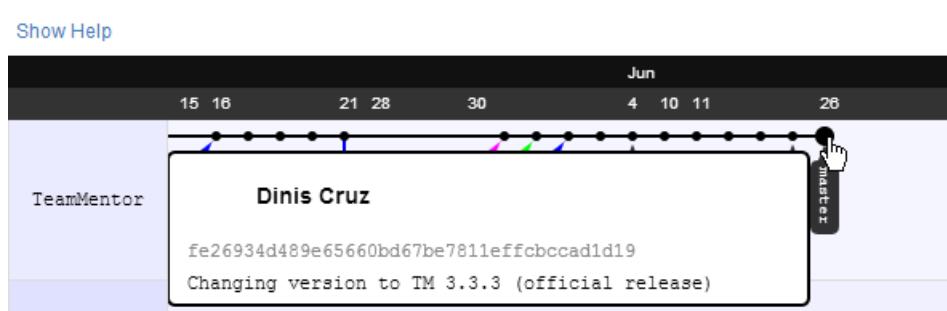
²<http://nvie.com/posts/a-successful-git-branching-model/>

³<http://blog.diniscruz.com/2013/05/great-presentation-on-git-branching.html>

⁴<https://github.com/TeamMentor/Master>

⁵<http://blog.diniscruz.com/2013/09/git-flow-moving-patches-from-one-commit.html>

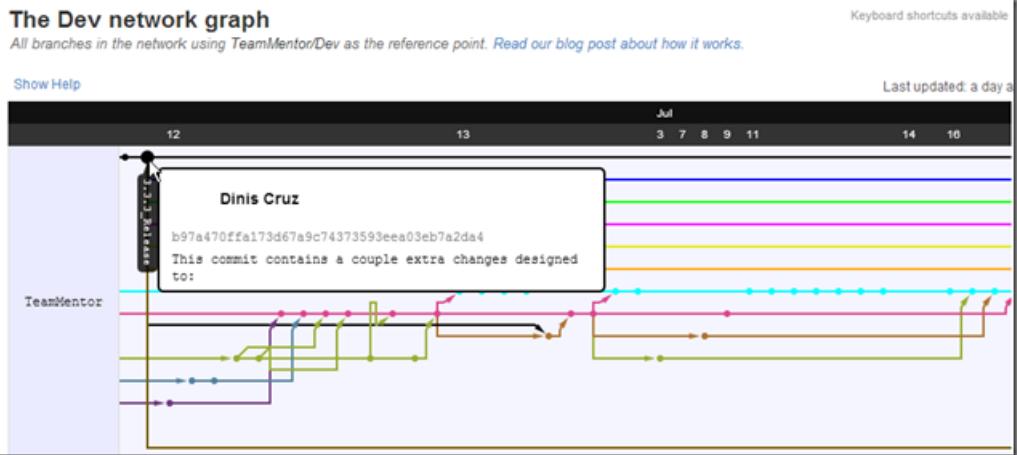
⁶<https://github.com/TeamMentor/Master>



.. and the 3_3_3_Hotfix branch on commit b97a470ffa173d67a9c74373593eea03eb7a2da4

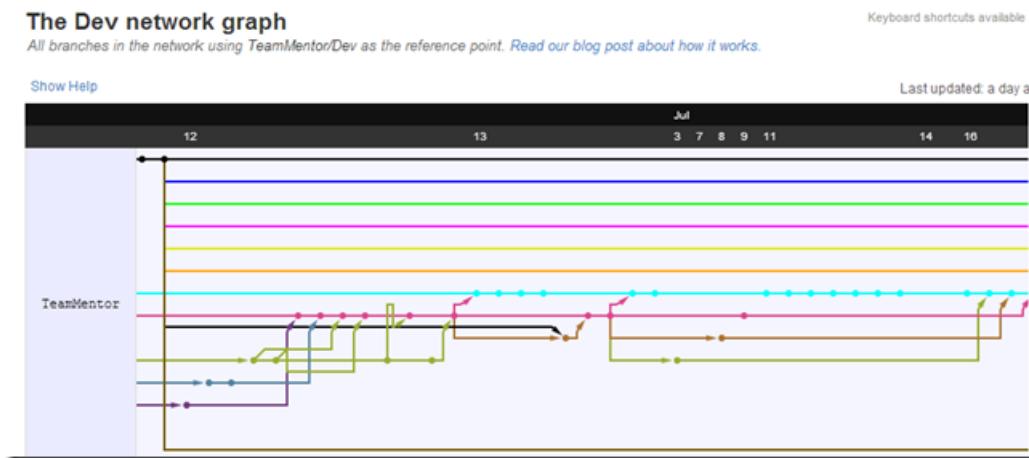


But looking at the [TeamMentor/Dev](#)⁷ Graph

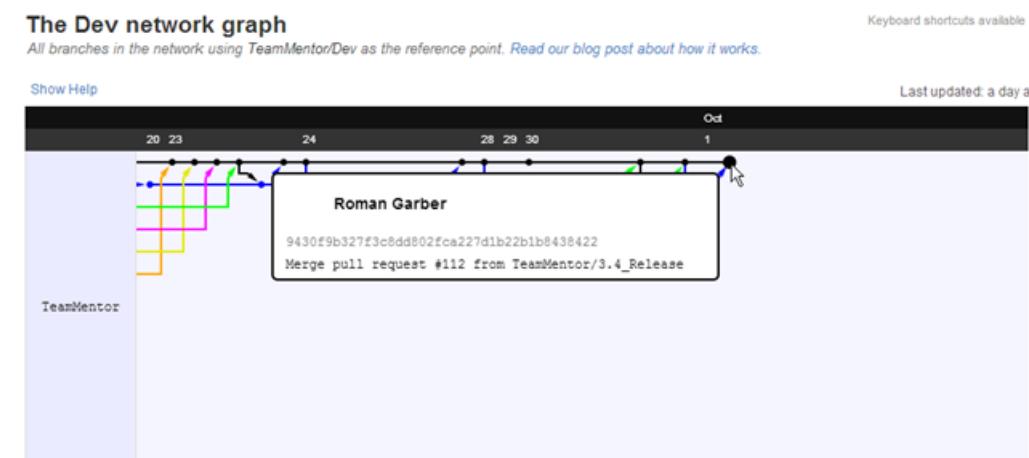


⁷<https://github.com/TeamMentor/Dev/>

...we can see that all commits (done on ‘one branch per issue’ workflow) have the **b97a470ffa173d67a9c743** commit as its parent (see image above and below)



In practice this means that the final 3.4 release commit from the [TeamMentor/Dev](#)⁸ repo



... is incompatible with the [TeamMentor/Master](#)⁹ repo (note that these could be branches of the same repo, but I like the use of separate repositories, since they provide a nice air-gap between development and production repositories)

⁸<https://github.com/TeamMentor/Dev/>

⁹<https://github.com/TeamMentor/Master>

Actually in principle they could be merged automatically if there was no conflicts!

But if we look at that extra commit from [TeamMentor/Master](#)¹⁰ repo (the [fe26934d489e65660bd67be7811effcbccad1d19](#) one)

https://github.com/TeamMentor/Master/commit/fe26934d489e65660bd67be7811effcbccad1d19

This repository ▾ Search or type a command Explore Gist Blog Help DinisCruz Unwatch 5 Star Unstar 7 Fork 11 PUBLIC TeamMentor / Master

Changing version to TM 3.3.3 (official release)

master v3.3.3

DinisCruz authored 3 months ago 1 parent b97a470 commit fe26934d489e65660bd67be7811effcbccad1d19

Showing 1 changed file with 1 addition and 1 deletion. [View file @ fe26934](#) [Show Diff Stats](#)

2 Web Applications/TM_Website/Javascript/TM/Settings.js

```

2 1 @@ -1,5 +1,5 @@
1 1 //TM Settings
2 2 +window.TM.tmVersion      = "TM 3.3.3 - Dev 1";
2 2 -window.TM.tmVersion      = "TM 3.3.3";
3 3 window.TM.ArticleTitle   = "TeamMentor 3.3";
4 4
5 5 window.TM.tmWebServices = '/Aspx_Pages/TM_WebServices.asmx/';

```

0 notes on commit fe26934 [Show line notes below](#)

... we see that the change was made on the version number (which in the 3.4 release will now say 3.4)

Note that GitHub will not allow a Pull Request to be made in cases like this, since GitHub has no online merge capabilities.

Ok, so how do we solve this?

The solution is to:

1. create a local branch pointing to [b97a470ffa173d67a9c74373593eea03eb7a2da4](#)
2. do a pull from [TeamMentor/Master](#)¹² to get the [fe26934d489e65660bd67be7811effcbccad1d19](#)¹³ commit
3. merge the current 3.4 code into [fe26934d489e65660bd67be7811effcbccad1d19](#)¹⁴ (which will cause a conflict)

¹⁰<https://github.com/TeamMentor/Master>

¹¹<https://github.com/TeamMentor/Master/commit/fe26934d489e65660bd67be7811effcbccad1d19>

¹²<https://github.com/TeamMentor/Master>

¹³<https://github.com/TeamMentor/Master/commit/fe26934d489e65660bd67be7811effcbccad1d19>

¹⁴<https://github.com/TeamMentor/Master/commit/fe26934d489e65660bd67be7811effcbccad1d19>

4. solve the conflict,
5. commit the result
6. push to GitHub into a new branch (called 3.4_Merge)
7. do a pull request (from 3.4_Merge into master)

In a local clone of [TeamMentor/Dev](#)¹⁵ we start by to create a branch that is pointing to b97a470ffa173d67a9c74373593eea03eb7a2da4

This can be done using the command: \$ git checkout b97a470ffa173d67a9c74373593eea03eb7a2da4

```
b2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <master>
$ git checkout b97a470ffa173d67a9c74373593eea03eb7a2da4
Note: checking out 'b97a470ffa173d67a9c74373593eea03eb7a2da4'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in thi
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so <now or later> by using -b with the checkout command again. Exampl
      git checkout -b new_branch_name

HEAD is now at b97a470... This commit contains a couple extra changes de
m/TeamMentor/Master/issues/547 issue - error handling on ip address ca
always good). Note: this fix is already on the 3.4 Dev branch - made t
t the moment new admin accounts (created if there is not one there) woul
t
b2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <(b97a470...)>
```

Followed by (as the help says) with: \$ git checkout -b 3.4_Merge

```
b2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <(b97a470...)>
$ git checkout -b 3.4_Merge
Switched to a new branch '3.4_Merge'
b2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <3.4_Merge>
```

Next we do a pull from TeamMentor/Master using \$ git pull git@github.com:TeamMentor/Master.git master:3.4_Merge

```
b2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <3.4_Merge>
$ git pull git@github.com:TeamMentor/Master.git master:3.4_Merge
From github.com:TeamMentor/Master
  b97a470..fe26934  master    -> 3.4_Merge
Warning: fetch updated the current branch head.
Warning: fast-forwarding your working tree from
Warning: commit b97a470ffa173d67a9c74373593eea03eb7a2da4.
Already up-to-date.
```

The command above is basically saying:

*Go to the git@github.com:TeamMentor/Master.git repo and merge/add the commits from its **master** branch into the local **3.4_Merge** branch*

¹⁵<https://github.com/TeamMentor/Dev/>

Note how the line `_b97a470..fe26934 master -> 3.4_Merge` (from screenshot above) shows how we went from the `b97a470ffa173d67a9c74373593eea03eb7a2da4` commit to the `**fe26934d489e65660bd67be7811effcbccad1d19 **commit`

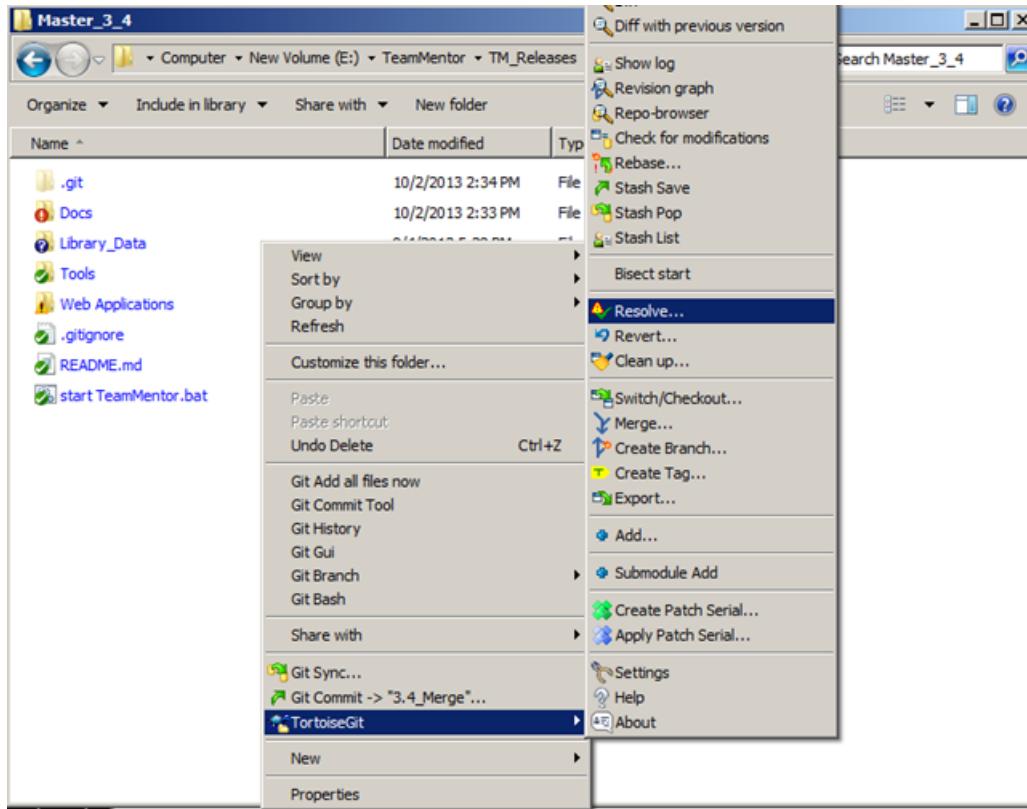
Next we merge into the `3.4_Merge` branch, the contents of the `master` branch (which contains the 3.4 code) using: `$ git merge master`

```
c:\200WIN\PCNQ5AABJ80 /E/TeamMentor/TM_Releases/Master_3_4 <3.4_Merge>
$ git merge master
Auto-merging Web Applications/TM_Website/Javascript/TM/Settings.js
CONFLICT (content): Merge conflict in Web Applications/TM_Website/Javascript/TM/Settings.js
Removing Web Applications/TM_Website/Javascript/TM_ControlPanel/TM.ControlPanel.WebServices.js
Removing Web Applications/TM_Website/Html_Pages/ControlPanel/WebServices/TM_WebServices.html
Auto-merging Docs/ReleaseNotes.v3.2.txt
Automatic merge failed; fix conflicts and then commit the result.
```

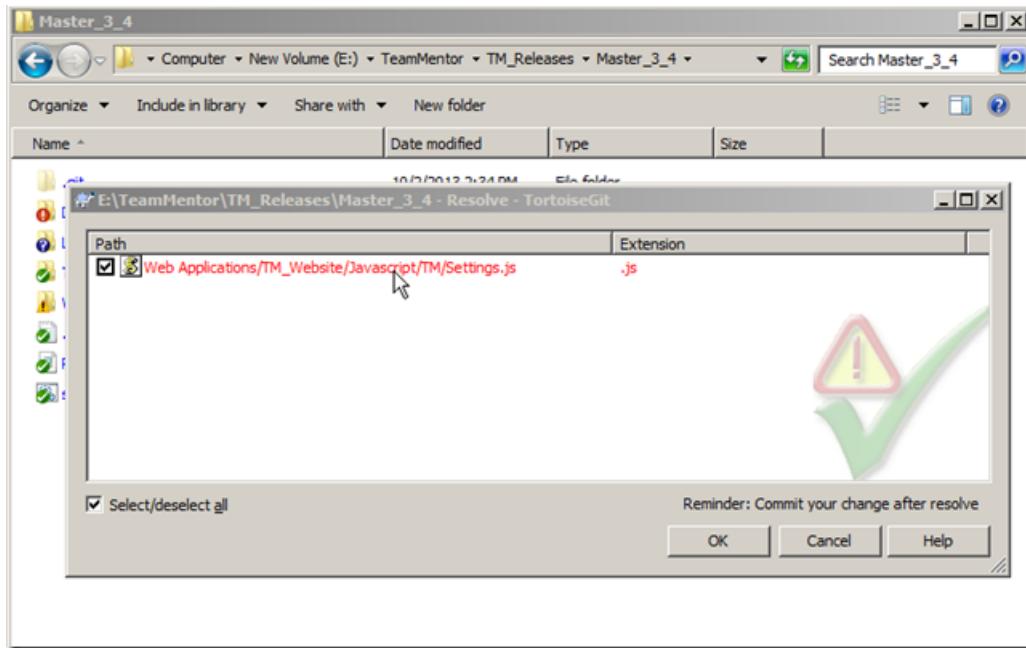
.... which predictably failed with a conflict on `Settings.js`

Solving git conflicts

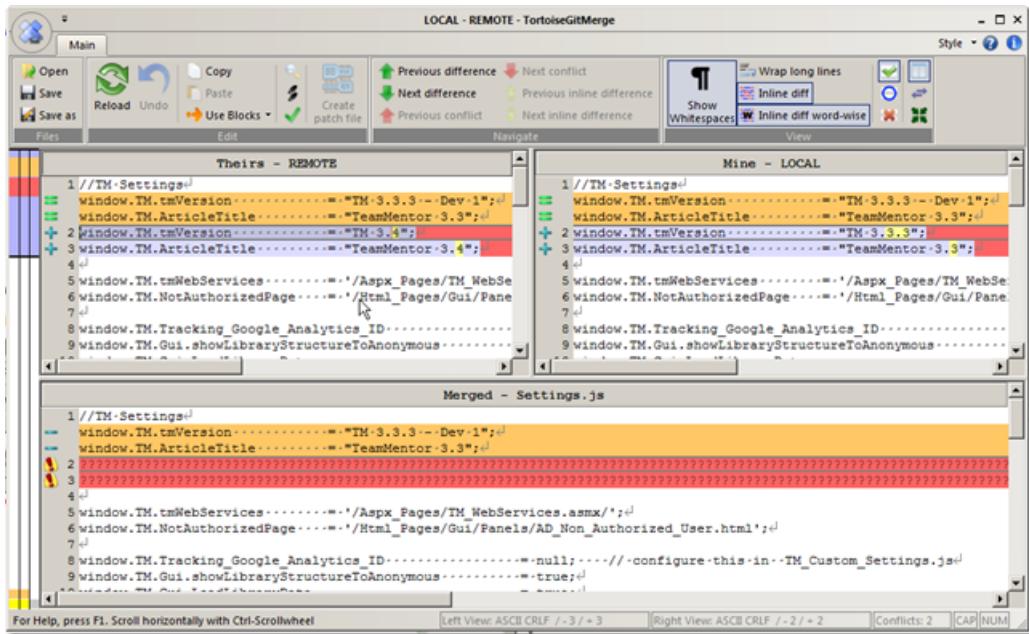
My preferred UI to solve conflicts is the one provided by TortoiseGit, which you can access from here:



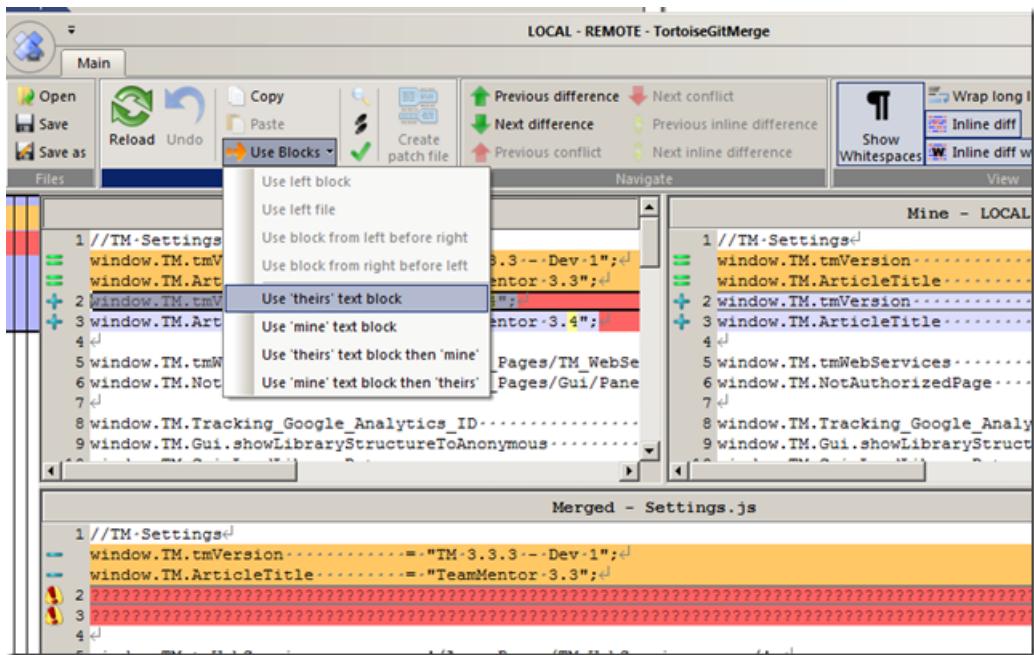
... them on the popup window that shows up, double click on the conflicted file:



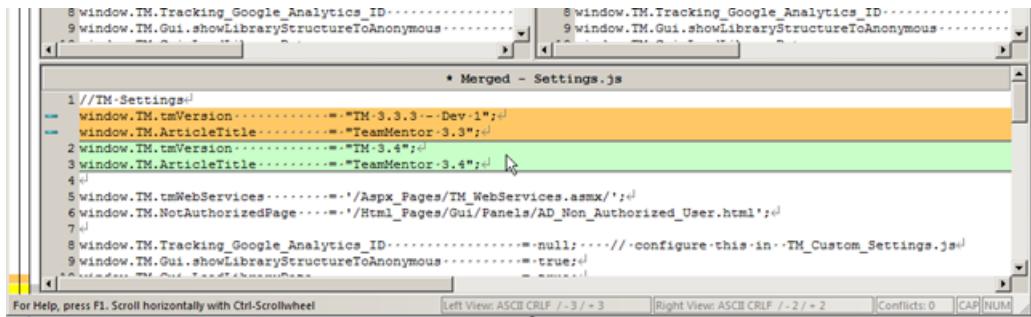
... and on the TortoiseMerge GUI :



... chose the option to *Use 'theirs' text block*



... which will update the bottom pane with the fixed version of Settings.js (in this case with no changes from before)

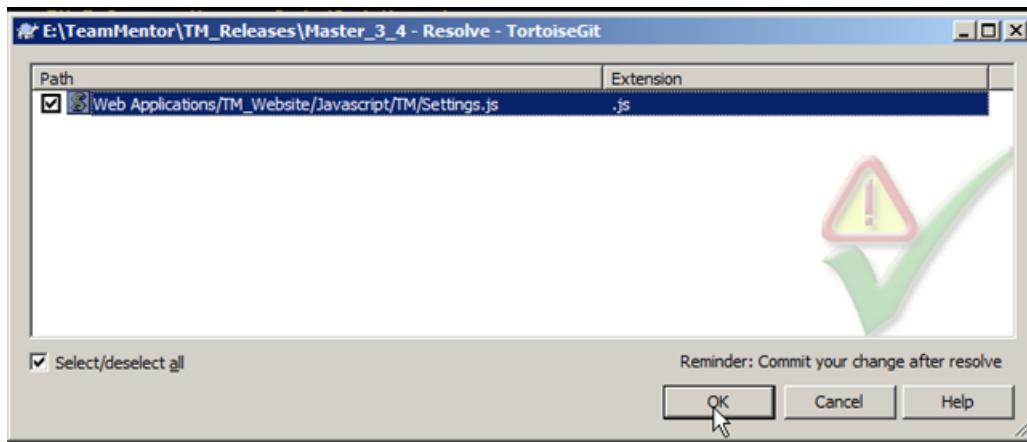


Save the changes and chose yes to mark the file as resolved:

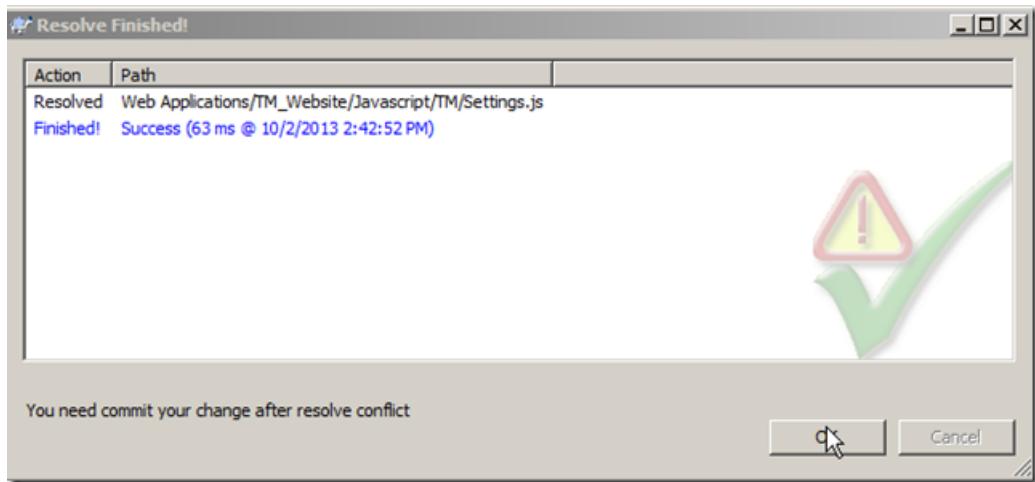
The screenshot shows a merge conflict between two versions of the file `Settings.js`. The left pane shows code from version 3.3, and the right pane shows code from version 3.4. A tooltip from the `TortoiseGitMerge` application asks if the user wants to mark the file as resolved. The "Yes" button is highlighted with a mouse cursor.

```
//TM-Settings:  
window.TM.tmVersion = "TM-3.3--Dev-1";  
window.TM.ArticleTitle = "TeamMentor-3.3";  
window.TM.ArticleTitle = "TeamMentor-3.4";  
window.TM.tmWebServices = "/Aspx_Pages/TM_WebServices.aspx/*";  
window.TM.NotAuthorizedPage = "/Html_Pages/Gui/Pages/Unauthorized.aspx";  
window.TM.Tracking_Google_Analytics_ID = "UA-XXXXXX-X";  
window.TM.Gui.showLibraryStructureToAnonymous();
```

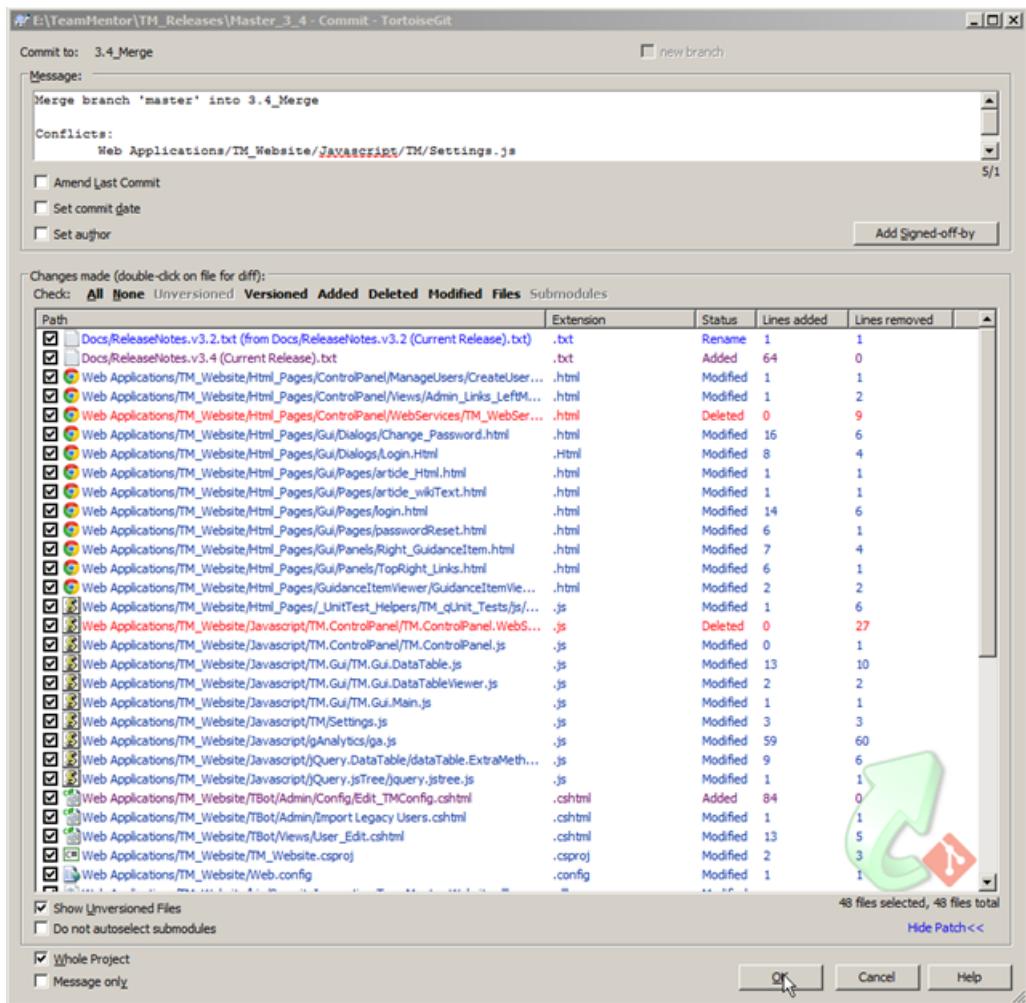
Close the TortoiseMerge and (since there is no other conflicts) click OK on the Resolve GUI



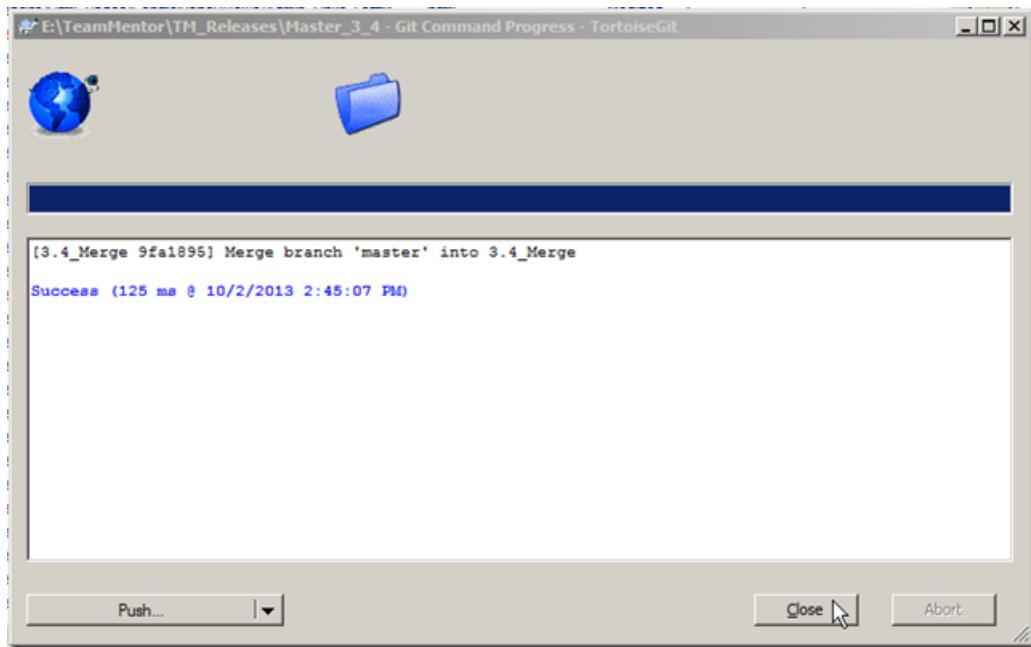
... another OK:



As the multiple ‘notes’ in the previous UIs mention, we need to commit the changes.
This commit will contain all changes including the conflict fixes



Once the commit is done:



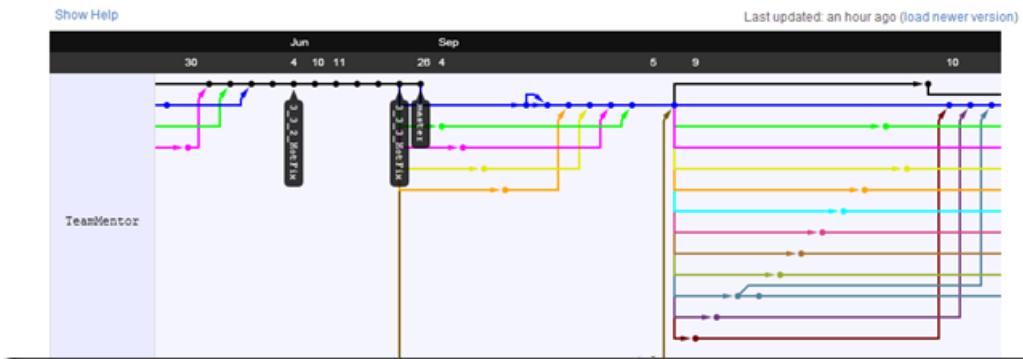
Go back to the Git Bash and push this branch into TeamMentor/Master (I prefer to do these things on a Git Bash)

```
o2@WIN-FGNQ5AARJ80 ~ /E/TeamMentor/TM_Releases/Master_3_4 <3.4_Merge>
$ git push git@github.com:TeamMentor/Master.git 3.4_Merge:3.4_Merge
Counting objects: 582, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (188/188), done.
Writing objects: 100% (489/489), 419.95 KiB / 0 bytes/s, done.
Total 489 (delta 394), reused 364 (delta 300)
To git@github.com:TeamMentor/Master.git
 * [new branch]      3.4_Merge -> 3.4_Merge
```

After the push, this is what the TeamMentor/Master graph looks like:

The Master network graph

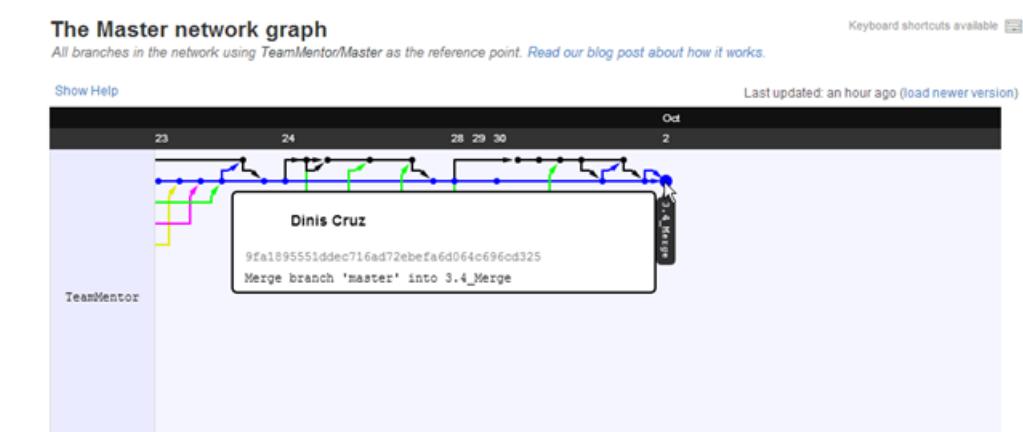
All branches in the network using TeamMentor/Master as the reference point. Read our blog post about how it works.



...with the 3.4 code now being there:

The Master network graph

All branches in the network using TeamMentor/Master as the reference point. Read our blog post about how it works.



Finally, what we can do now is to issue a Pull Request:

This screenshot shows the GitHub pull request interface for the 'TeamMentor / Master' repository. The top navigation bar shows the repository name, watch/fork counts, and a 'New pull request' button. Below the header, a search bar and a 'Yours' section are visible. The main area displays a message stating 'No pull requests to show.' A green 'New pull request' button is prominently displayed in the top right corner of the main content area.

... from the _ 3.4_Merge_ branch:

The screenshot shows a GitHub repository page for 'TeamMentor / Master'. At the top, there are buttons for 'Unwatch' (5), 'Unstar' (7), and 'Fork' (11). Below the header, a message says 'We're showing branches in this repository, but you can also compare across forks.' with dropdown menus for 'base: master' and 'compare: master'. A large button labeled 'Compare branches to draft a pull request.' is visible. The main area features a section titled 'Compare and review just about anything.' with icons for branches, tags, and commit ranges. Below this, a table lists 'EXAMPLE COMPARISONS':

	3.4_Merge	3 minutes ago
	3_1_Release	2 years ago
	master@{1day}...master	24 hours ago

... into the master branch:

The screenshot shows the same GitHub repository page for 'TeamMentor / Master'. The top navigation bar and fork/follow buttons are identical. The main content area now displays the details of the 'master' branch. It shows 86 commits, 48 files changed, 0 comments, and 4 contributors. Below this, tabs for 'Commits', 'Files Changed', and 'Commit Comments' are visible. The 'Commits' tab is selected, showing a list of recent commits from Sep 04, 2013:

Date	Author	Message	Hash
Sep 04, 2013	michaelhid...	Fixing Issue_142 ...	5319e30
Sep 04, 2013	michaelhid...	Fixing Issue 51 ...	c9d5cb9

... which contain all the code changes since the 3.3.3 release

TeamMentor / Master

3.4 merge

Write Preview Comments are parsed with GitHub Flavored Markdown

This is the 3.4 changes since 3.3.3 release

Able to merge

These branches can be automatically merged

Send pull request

With the best part being that this Pull Request can be merged using GitHub's UI (since there are no conflicts)

roman87 Quick fix of the release notes 67e9cdc

roman87 Merge pull request #112 from TeamMentor/3.4_Release 9430f9b

DinisCruz Merge branch 'master' into 3.4_Merge 9fa1895

DinisCruz Merge branch 'master' into 3.4_Merge 8eb3df7

You can add more commits to this pull request by pushing to the 3.4_Merge branch on TeamMentor/Master

This pull request can be automatically merged. You can also merge branches on the command line.

Merge pull request

Leave a comment

Tip: You can also add notes to lines changed in a file under Diff

Comment

And that's it!

Hopefully this provided a good example of how to use Git and TortoiseGit to easily merge commits and resolve any resulting conflicts.

Tip: How to delete branches in GitHub:

To delete a branch in Github, we do a push from an 'empty branch' into an 'existing branch'

In this case, if I wanted to delete the 3.4_Merge branch at the TeamMentor/Master repository, I would use: \$ git push git@github.com:TeamMentor/Master.git :3.4_Merge

```
$ git push git@github.com:TeamMentor/Master.git :3.4_Merge
To git@github.com:TeamMentor/Master.git
 - [deleted]      3.4_Merge
```

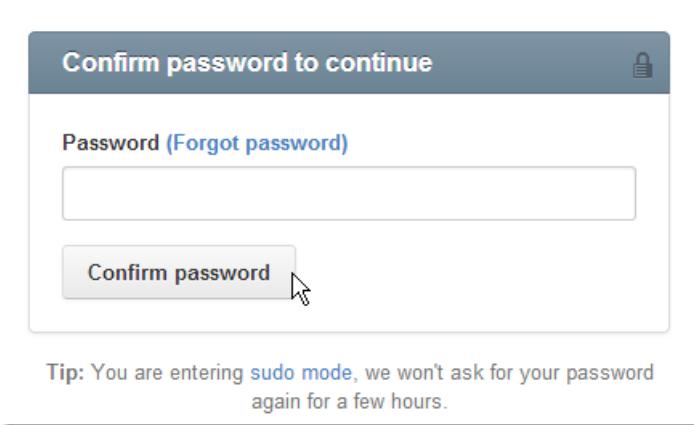
Enabling GitHub Two Factor Authentication

Inspired by Google's Two Factor Authentication workflow, last month GitHub did the same thing.

I just enabled it, and I strongly recommend that you do it to.

As per the instructions in GitHub's [Two-factor Authentication¹⁶](#) post, the first step is to go to [A screenshot of a web browser showing the GitHub Two-factor authentication settings page. The title bar says 'Two-factor authentication'. Below it, a status message says 'Status: Off' with a red 'X'. A large button labeled 'Set up two-factor authentication' is centered. Below the button, a note explains that two-factor authentication provides another layer of security and links to 'GitHub Help'.](https://github.com/settings/admin¹⁷ and click on the <i>'Set up two-factor authentication'</i> button:</p></div><div data-bbox=)

... which requires the current password to be entered:



In this case I'm going to use SMS:

¹⁶<https://github.com/blog/1614-two-factor-authentication>

¹⁷<https://github.com/settings/admin>

Two-factor authentication adds an extra layer of security to your account. In addition to your username and password, you'll need to enter a code that GitHub sends to you via text or an app on your phone.



When you sign into GitHub you'll enter your username and password, like always.

When logging in from a new device, you'll need to enter an additional code sent to your phone or tablet to verify your identity.

Once you enter the code on the website, you'll be logged into GitHub.

[Set up using SMS](#)

[Set up using an app](#)

Next we enter the phone number and click on *Send code*:

Enable two-factor authentication via SMS

1. Provide a phone number where you can receive SMS messages

Country code

Phone number

2. Enter the 6-digit code we sent to you

[Back to settings](#)

... enter the number received by SMS and click **Enable**

2. Enter the 6-digit code we sent to you

And that's it, two-factor authentication is now enabled:

The screenshot shows the GitHub Two-factor Authentication settings page. At the top, a blue banner displays the message "Two-factor authentication successfully enabled!". Below this, the "Disable two-factor authentication" section is shown, indicating that two-factor authentication is currently "On" with a green checkmark. A red button labeled "Disable two-factor authentication" is present. A note below states: "You'll only need your username and password to login after disabling two-factor authentication." The next section, "Recovery codes", explains that recovery codes allow access if the primary device is lost. It includes a link to "View recovery codes" and a link to "Download recovery codes as .txt file". The final section, "Primary two-factor authentication code delivery options", shows that the primary method is "SMS message", recommended for adding a fallback number. It notes that SMS delivery is configured for a number ending in 557. A red "Reconfigure two-factor authentication" button is at the bottom of this section.

Creating Tokens to access repos (instead of pwds)

Also great from a security point of view, is that it is also possible to create 'login tokens' for Https logins.

This is done one the *Applications* Settings page:

The screenshot shows the GitHub settings page for 'DinisCruz'. On the left is a sidebar with links: Profile, Account Settings, Emails, Notification Center, Billing, Payment History, and SSH Keys. The main area is titled 'Developer applications' with a link to 'Register new application'. Below that is the 'Personal Access Tokens' section. A message at the top says: 'Make sure to copy your new personal access token now. You won't be able to see it again!' A button labeled 'Create new token' is visible. A note below explains: 'Personal Access Tokens function like ordinary OAuth access tokens. They can be used instead of a password for git over HTTPS, or can be used to authenticate to the API over Basic Authentication.' A table lists tokens, with one entry: 'iOctocat - GitHub client for iOS – Edit' and a 'Delete' button.

... where new tokens can be created:

The screenshot shows a modal dialog titled 'Create a new Personal Access Token'. It contains a note about what Personal Access Tokens are, a field for 'Enter a name for the token below so you can keep track of where you used it and revoke the token if necessary.', and a 'Token description:' field containing 'test " > token'. A 'Create Token' button is at the bottom right, with a cursor pointing at it.

... which can now be used instead of passwords (with the great advantage of being revocable and assignable for a particular use (lets say a particular deployment or app))

The screenshot shows the 'Personal Access Tokens' list. It includes a note about what Personal Access Tokens are, a table listing tokens, and a 'Create new token' button. One token is listed: '60396b54a6fb112eb0cc6ffbdaf7f936c2b34bcc' with a description 'test " > token – Edit' and a 'Delete' button. Another token, 'iOctocat - GitHub client for iOS – Edit', also has a 'Delete' button.

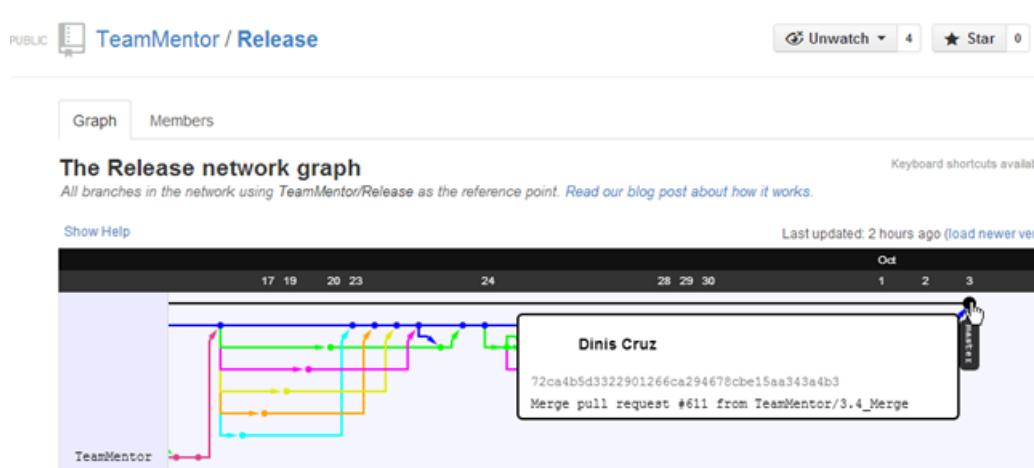
I really like this functionality, and hope to eventually add something similar to TeamMentor

Syncing all releases to the same commit and Tag (for TeamMentor v3.4)

This is a bit of house keeping, as you can see by the [Fixing the Merge conflict caused by one extra commit on TeamMentor master¹⁸](#) and [Git Flow - Moving patches from one Commit into another Commit¹⁹](#) posts, not doing this has already cause us some pain in the past.

So after some pushes and pulls (of both commits and tags) I now have the main TeamMentor repos all synchronized at the [72ca4b5d3322901266ca294678cbe15aa343a4b3²⁰](#) commit:

[TeamMentor/Release²¹](#) - now the new official home for TeamMentor releases (i.e. the ‘production code’)



[TeamMentor/Master²²](#) – the old TeamMentor official repo (and updated to help migrations into 3.4)

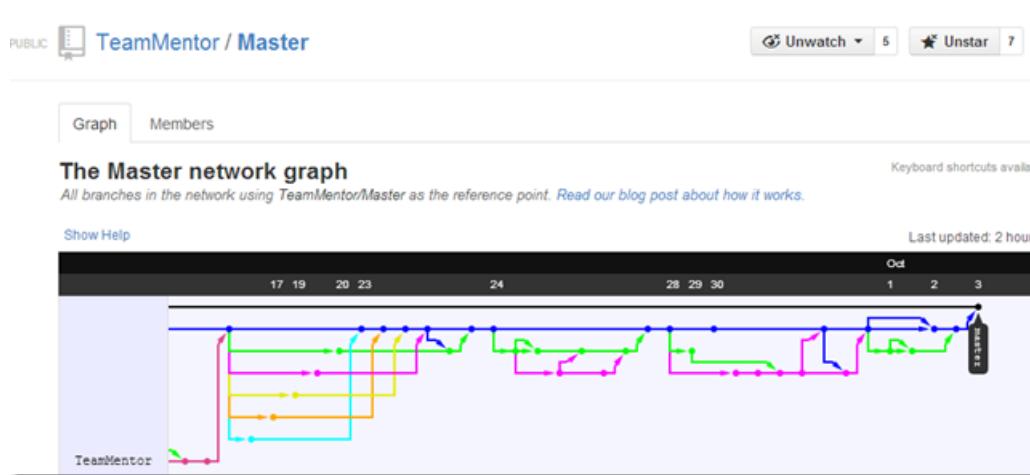
¹⁸<http://blog.diniscruz.com/2013/10/fixing-merge-conflict-caused-by-one.html>

¹⁹<http://blog.diniscruz.com/2013/09/git-flow-moving-patches-from-one-commit.html>

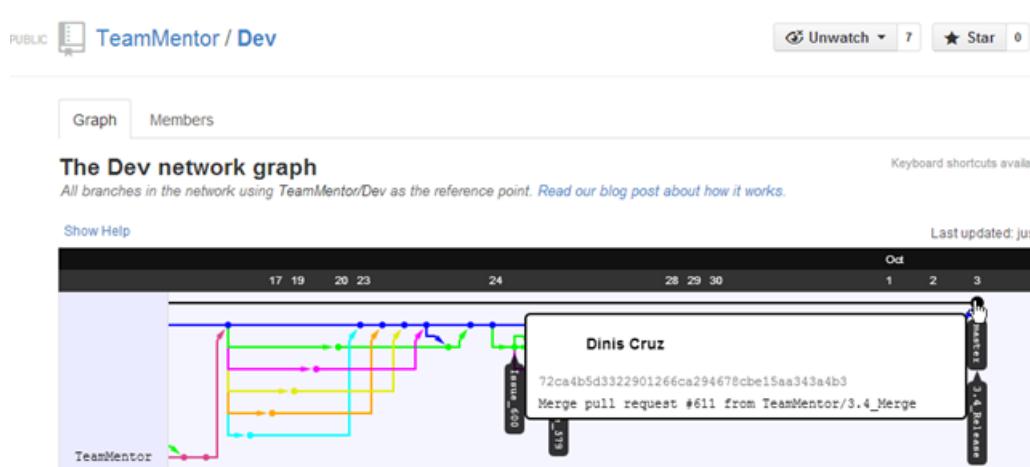
²⁰<https://github.com/TeamMentor/Release/commit/72ca4b5d3322901266ca294678cbe15aa343a4b3>

²¹<https://github.com/TeamMentor/Release>

²²<https://github.com/TeamMentor/Master/>



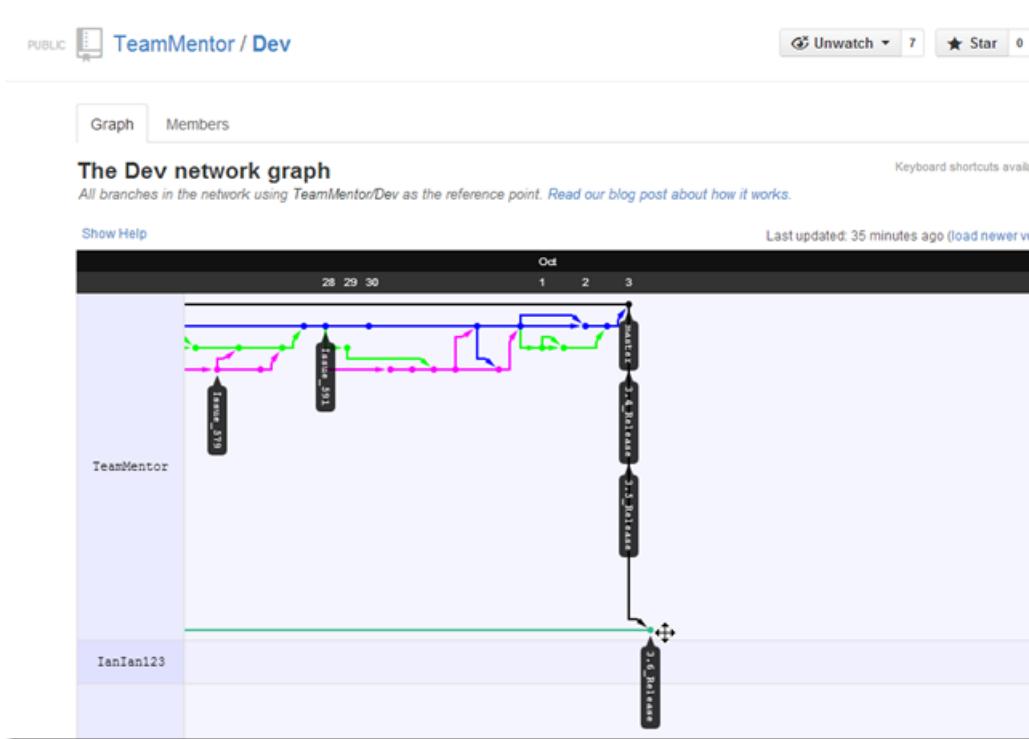
TeamMentor/Dev²³ #1 (before merge 3.5_Release and 3.6_Release merges) – this is the main development clone/fork



TeamMentor/Dev²⁴ #2 (after merge 3.5_Release and 3.6_Release merges) – note how 3.5 is currently at the same level as 3.4 (see Appendix 1 and Appendix 2 for how this was done, and how the merge conflicts were resolved)

²³<https://github.com/TeamMentor/De>

²⁴<https://github.com/TeamMentor/De>



This can be further confirmed by GitHub's Branches view, where the ***3.5_Release*** and ***3.4_Release*** branches are synced with master:

The screenshot shows the GitHub 'Branches' page for the 'TeamMentor / Dev' repository. It lists the following branches:

- master**: Last updated 3 hours ago by DinisCruz. This is the base branch.
- 3.5_Release**: Last updated 3 hours ago by DinisCruz. 0 ahead, 0 behind.
- 3.4_Release**: Last updated 3 hours ago by DinisCruz. 0 ahead, 0 behind.
- 3.3.3_Release**: Last updated 4 months ago by DinisCruz. 0 ahead, 88 behind.

Each branch row includes 'Delete branch' and 'Compare' buttons.

... and the ***3.6_Release*** branch is already 131 commits ahead of ***master*** (and ***3.4_Release*** and ***3.5_Release***)

The screenshot shows the GitHub interface for the repository 'TeamMentor / Dev'. At the top, there are buttons for 'Unwatch' (with 7 notifications), 'Star' (0 stars), and 'Follow'. Below that, the repository name 'TeamMentor / Dev' is displayed. The main section is titled 'Branches' and shows 'Showing 2 branches not merged into master. View merged branches.' There are three branches listed:

- master**: Last updated 3 hours ago by [DinisCruz](#). This is the 'Base branch'.
- 3.6_Release**: Last updated 8 minutes ago by [DinisCruz](#). It is ahead by 131 commits.
- NodeJS**: Last updated 8 months ago by [DinisCruz](#). It is behind by 310 commits.

For each branch, there are 'Delete branch' and 'Compare' buttons.

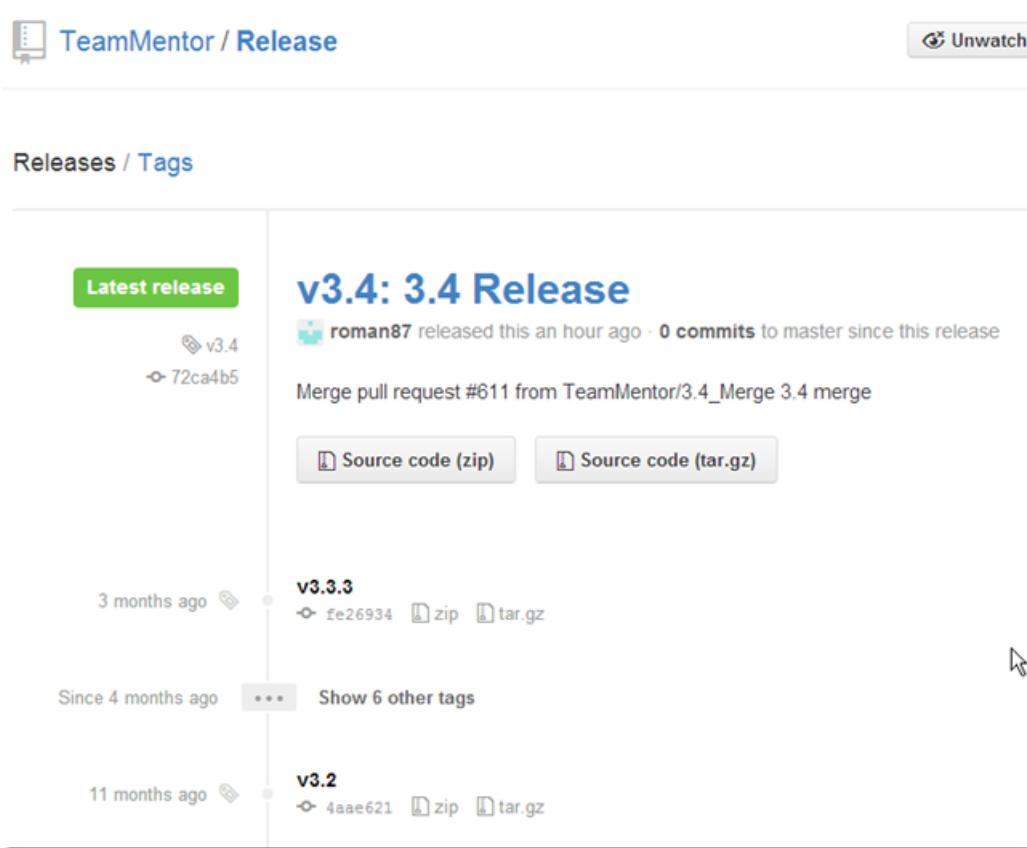
Updating the tags/Releases

I also updated the tags of 3 repos (Release, Master, Dev).

NOTE:** I had to force the update of the tags**, since there was already an v3.4 tag in there (I'm not 100% sure of the side effects of this, but I'm sure I'll soon find out :))

Release repo (new production release): <https://github.com/TeamMentor/Release/releases>²⁵

²⁵<https://github.com/TeamMentor/Release/releases>



The screenshot shows the GitHub releases page for the repository `TeamMentor/Master`. The main heading is `v3.4: 3.4 Release`, which is highlighted as the `Latest release`. Below the heading, it says `roman87 released this an hour ago · 0 commits to master since this release`. A note indicates a `Merge pull request #611 from TeamMentor/3.4_Merge 3.4 merge`. There are two download buttons: `Source code (zip)` and `Source code (tar.gz)`. The release was made `3 months ago`. Below this, there's a section for other tags, showing `v3.3.3` (released `3 months ago`) and `v3.2` (released `11 months ago`). A link to `Show 6 other tags` is available. At the bottom of the page, there's a link to `https://github.com/TeamMentor/Master/releases`.

Master repo (legacy/previous production release) [https://github.com/TeamMentor/Master/releases²⁶](https://github.com/TeamMentor/Master/releases)
(note sure why the GitHub's layout of this one is different)

²⁶<https://github.com/TeamMentor/Master/releases>

The screenshot shows the GitHub releases page for the 'TeamMentor / Master' repository. At the top, there is a 'Unwatch' button with a dropdown menu showing '5'. Below the header, the title 'Releases / Tags' is displayed. A horizontal timeline lists six releases from 'an hour ago' to '10 months ago'. Each release entry includes the tag name, a copy icon, a commit hash, and download links for 'zip' and 'tar.gz'.

Released	Tag	Commit Hash	Downloads
an hour ago	v3.4	72ca4b5	zip tar.gz
3 months ago	v3.3.3	fe26934	zip tar.gz
4 months ago	v3.3.2	6f63a57	zip tar.gz
5 months ago	v3.3.1	a9eb167	zip tar.gz
6 months ago	v3.3	e004e1d	zip tar.gz
10 months ago	v3.2.4	4e39135	zip tar.gz

Dev repo: [https://github.com/TeamMentor/Dev/releases²⁷](https://github.com/TeamMentor/Dev/releases)

²⁷<https://github.com/TeamMentor/Dev/releases>

The screenshot shows the GitHub releases page for the `TeamMentor / Dev` branch. At the top, there's a green button labeled "Latest release". Below it, the latest release is listed as **v3.4: 3.4 Release**, released an hour ago by `roman87`. It includes a merge pull request from `TeamMentor/3.4_Release`. There are two download links: "Source code (zip)" and "Source code (tar.gz)".

Below this, there are other releases:

- v3.3.3**, released a month ago by `b97a470`. It has download links for "zip" and "tar.gz".
- v3.2**, released 11 months ago by `4aae621`. It has download links for "zip" and "tar.gz".

A link "Show 5 other tags" is available to view more releases.

Appendix 1: Updating 3.5 Release branch

This one was easy since the `3.5_Release` branch was already synced with an earlier version of the `3.4_Release` branch:

```

$ git checkout 3.5_Release
Branch 3.5_Release set up to track remote branch '3.5_Release'.
Switched to a new branch '3.5_Release'
o2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <3.5_Release>
$ git merge master
Updating 48c0ec0..72ca4b5
Fast-forward
  ...<Current Release>.txt => ReleaseNotes.v3.2.txt      |  2 ++
  Docs/ReleaseNotes.v3.4 <Current Release>.txt          | 64 ++++++=====
  .../ControlPanel/ManageUsers/CreateUsers.html           |  2 ++
  .../Html_Pages/Gui/Pages/article_Html.html            |  2 ++
  .../Html_Pages/Gui/Pages/article_wikiText.html        |  2 ++
  .../TM_Website/Html_Pages/Gui/Pages/login.html       | 20 +++-
  .../Html_Pages/Gui/Pages/passwordReset.html           |  7 ++
  .../Html_Pages/Gui/Panels/Right_GuidanceItem.html     |  8 ++
  .../GuidanceItemViewer/GuidanceItemViewer.html        |  2 ++
  .../Javascript/TM.Gui/TM.Gui.DataTable.js             |  2 ++
  .../TM_Website/Javascript/TM/Settings.js              |  4 ++
  .../TM_Website/Javascript/gAnalytics/ga.js           | 24 +***-
  .../IBot/Admin/Config/Edit_IMConfig.cshtml           | 84 ++++++=====
  .../TM_Website/TBot/Views/User_Edit.cshtml           | 14 +++-
  Web Applications/TM_Website/TM_Website.csproj         |  3 ++
  Web Applications/TM_Website/Web.config                |  2 ++
  .../bin/SecurityInnovation.TeamMentor.Website.dll    | Bin 4608 -> 4608 bytes
  .../TM_Website/bin/TeamMentor.CoreLib.dll             | Bin 346751 -> 348290 bytes
  .../TM_AppCode/Authentication/TM_Authentication.cs   |  5 ++
  .../TM_AppCode/Schemas/Models/TM_User.cs              |  2 ++
  .../TeamMentor.CoreLib/TM_AppCode/TMConsts.cs         | 22 +*****
  .../TM_AppCode/Users/TMUser_ExtensionMethods.cs       | 23 +*****+
  .../TM_AppCode/Utils/SendEmails.cs                   | 16 +***-
  .../WebServices/TM_Wbservices.Config.cs               |  8 ++
  .../WebServices/TM_Wbservices.Session.cs              |  2 ++
  .../XmlDatabase/TM_Xml_Database.Library.cs           |  7 ++
  .../Asmx_WebServices/Test_Libraries.cs                | 20 +***-
  .../Test_VIEWS_Folders_Articles.cs                  | 25 +*****+
  .../CoreLib/Test_HandleRequest_LibraryData.cs        |  4 ++
  .../TeamMentor.UnitTesting/CoreLib/Test_Schemas.cs    |  4 ++
  .../TeamMentor.UnitTesting/Messaging/Test_Email.cs    | 32 +*****+
  .../REST_Direct/Test_REST_Admin.cs                  |  3 ++
  .../TM_XmlDatabase/Test_Library_SaveToDisk.cs       |  2 ++
  Web Applications/build.bat                          |  1 +
34 files changed, 345 insertions(+), 73 deletions(-)
rename Docs/ReleaseNotes.v3.2 <Current Release>.txt => ReleaseNotes.v3.2.txt (94%)
create mode 100644 Docs/ReleaseNotes.v3.4 <Current Release>.txt
create mode 100644 Web Applications/TM_Website/IBot/Admin/Config/Edit_IMConfig.cshtml
create mode 100644 Web Applications/build.bat

```

The push also confirms that this branch is at 72ca4b5d3322901266ca294678cbe15aa343a4b3

```

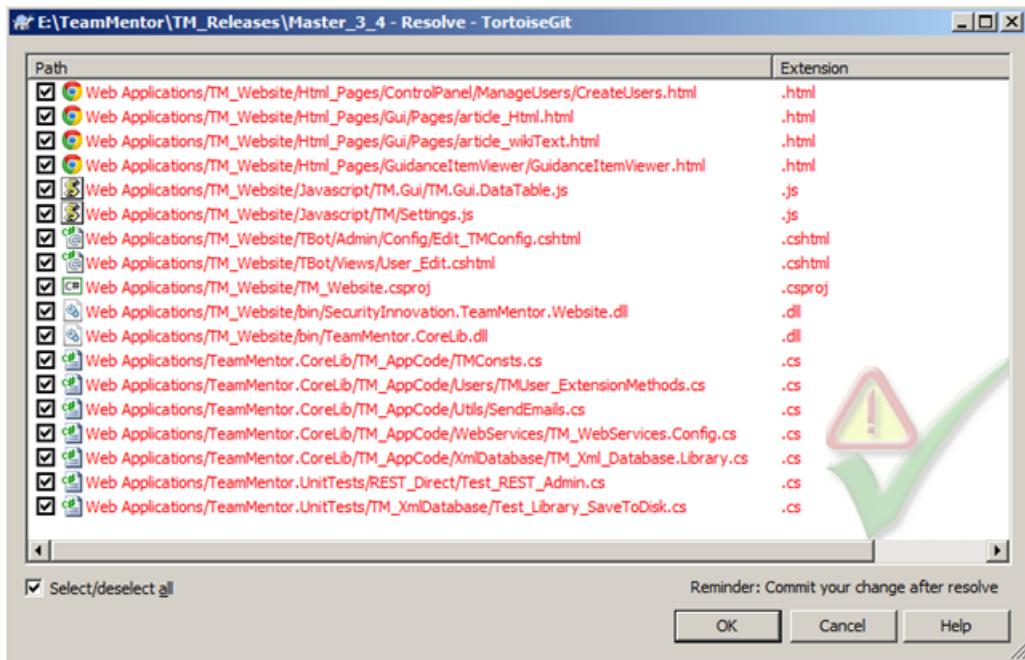
o2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <3.5_Release>
$ git push dev 3.5_Release:3.5_Release
Total 0 <delta 0>, reused 0 <delta 0>
To git@github.com:TeamMentor/Dev.git
  48c0ec0..72ca4b5  3.5_Release -> 3.5_Release

```

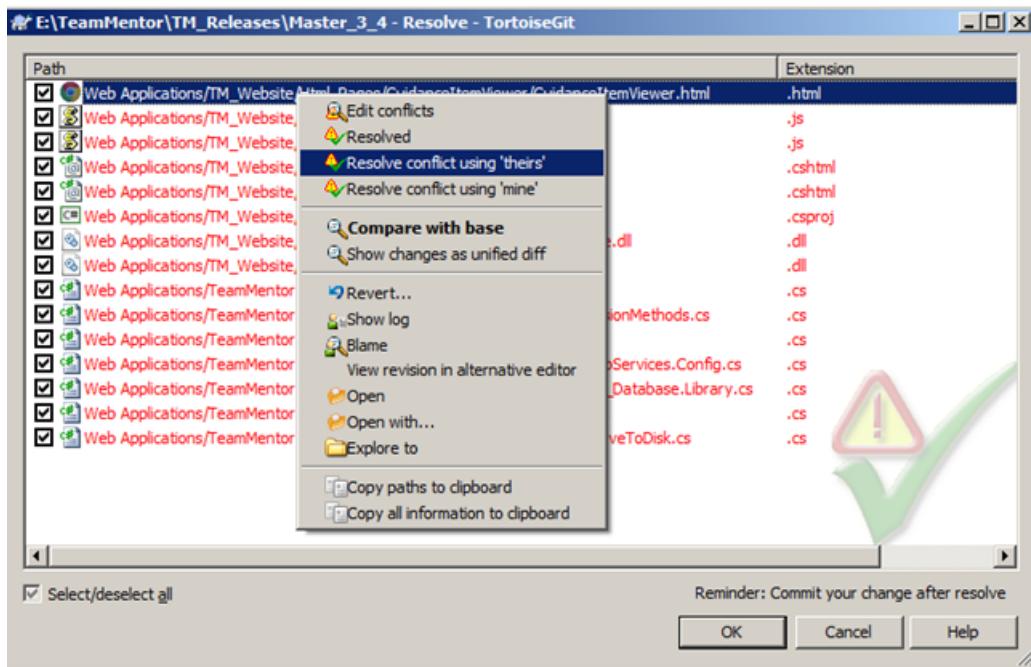
Appendix 2: Updating 3.6 Release branch

Merging the 3.4_Release into 3.6_Release was a bit more problematic due to the number of changes/fixes already done on 3.6 and the backporting of some fixes to 3.4.

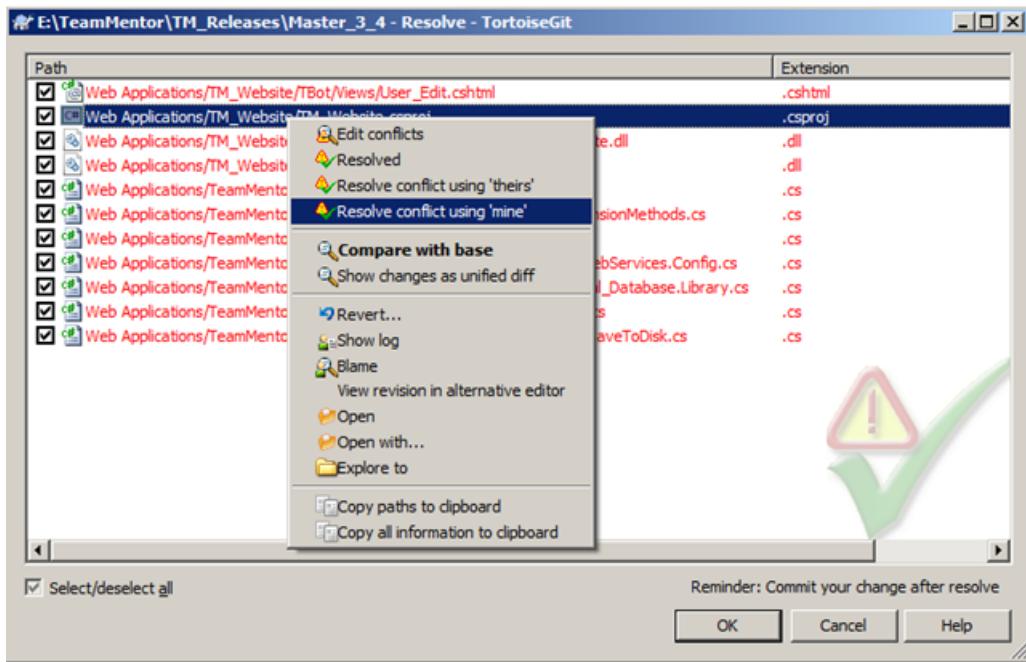
After the merge failed, here are the conflicts that needed to be solved:



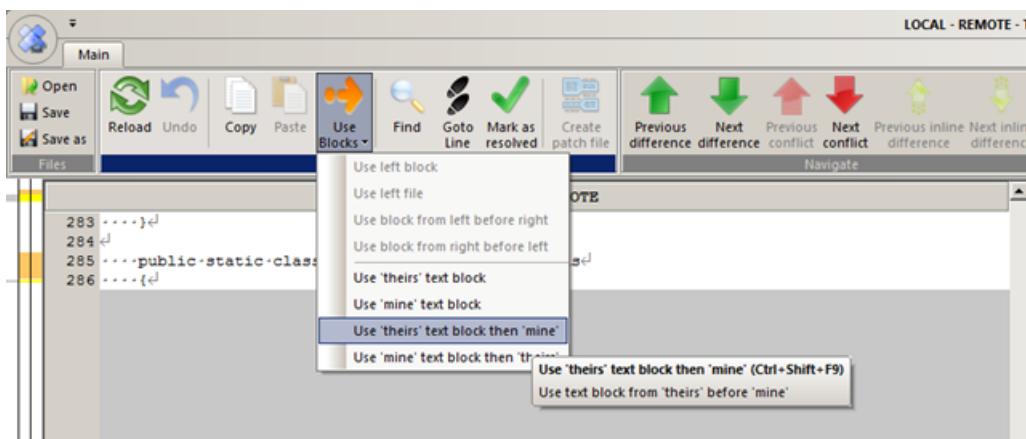
... with some resolved using the 'theirs' strategy



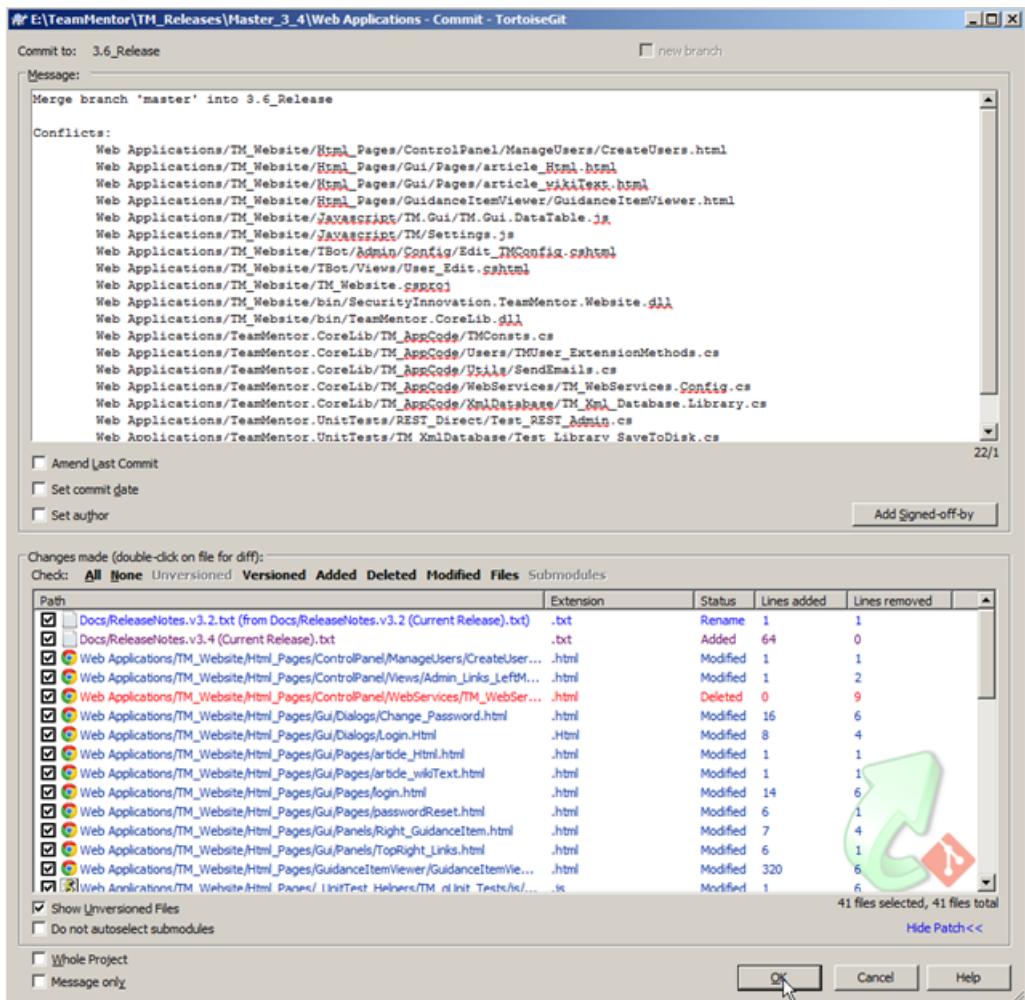
... a few using the '*mine*' strategy



... and a couple using the '*theirs first then mine*' strategy (which I expect that will need further fixing in VisualStudio)



Finally here is the commit that applies all 3.4 changes to 3.6 (including the merge fixes)



... and the respective push:

```
n2@WIN-FGNQ5AARJ80 /E/TeamMentor/TM_Releases/Master_3_4 <3.6_Release>
$ git push dev 3.6_Release:3.6_Release
Counting objects: 148, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (49/49), done.
Writing objects: 100% (50/50), 8.28 KiB | 0 bytes/s, done.
Total 50 (delta 45), reused 0 (delta 0)
To git@github.com:TeamMentor/Dev.git
  16354b3..8b45912 3.6_Release -> 3.6_Release
```

12. January 2014

- How to update a forked GitHub repo (in this case tm-sme/Lib_Vulnerabilities)
- Updating the GitHub repos for the 1.6.0 release of the Eclipse Fortify Plugin
- Updating GitHub Forks with latest commits from GitHub's 'parent' repo
- Using TeamMentor 3.4 TBot admin pages to load and sync a Library hosted on GitHub
- Using TeamMentor 3.4 TBot admin pages to load and sync UserData with a GitHub hosted repo
- Adding files to TeamMentor's web root via a UserData folder (synced with GitHub)

How to update a forked GitHub repo (in this case tm-sme/Lib_Vulnerabilities)

Today I helped to update the [tm-sme/Lib_Vulnerabilities¹](https://github.com/tm-sme/Lib_Vulnerabilities) repo which is a fork of the [TMContent/Lib_Vulnerabilities²](https://github.com/TMContent/Lib_Vulnerabilities) and is being auto-updated in real-time when changes made to the [https://sme.teammentor.net/³](https://sme.teammentor.net/) server (i.e every time there is a content change in [https://sme.teammentor.net/⁴](https://sme.teammentor.net/) there is a server-side *git commit*, followed by a *git pull* to [tm-sme/Lib_Vulnerabilities⁵](https://github.com/tm-sme/Lib_Vulnerabilities) (which is a pretty sweet workflow))

The issue we had was how to push the changes from [tm-sme/Lib_Vulnerabilities⁶](https://github.com/tm-sme/Lib_Vulnerabilities) into the [TMContent/Lib_Vulnerabilities⁷](https://github.com/TMContent/Lib_Vulnerabilities) repo, so that they can be synced back to [https://vulnerabilities.teammentor.net/⁸](https://vulnerabilities.teammentor.net/)

Note: this workflow would have been easier if the two repos were in sync, but it happened that there was one commit made to [TMContent/Lib_Vulnerabilities⁹](https://github.com/TMContent/Lib_Vulnerabilities) (which is the master repo) on the [13th of Dec \(d26f385\)¹⁰](#) in between a bunch of updates to the [tm-sme/Lib_Vulnerabilities¹¹](https://github.com/tm-sme/Lib_Vulnerabilities) repo (done automatically by TeamMentor). Bottom line: at this stage the repos are not compatible, which is why the GitHub Pull Requests don't work.

Here are the Git commands I executed locally to merge these repos successfully:

Step 1) Clone repo and try to do a simple pull

- 1) git plugin\$ git clone git@github.com:tm-sme/Lib_Vulnerabilities.git
 - 2) cd Lib_Vulnerabilities/
 - 3) git remote add upstream git@github.com:TMContent/Lib_Vulnerabilities.git
 - 4) git checkout -b mergeBranch
 - 5) git pull upstream master:mergeBranch
- which doesn't work:

```
! [rejected] master -> mergeBranch (non-fast-forward)_
```

¹https://github.com/tm-sme/Lib_Vulnerabilities

²https://github.com/TMContent/Lib_Vulnerabilities

³<https://sme.teammentor.net/>

⁴<https://sme.teammentor.net/>

⁵https://github.com/tm-sme/Lib_Vulnerabilities

⁶https://github.com/tm-sme/Lib_Vulnerabilities

⁷https://github.com/TMContent/Lib_Vulnerabilities

⁸<https://vulnerabilities.teammentor.net/>

⁹https://github.com/TMContent/Lib_Vulnerabilities

¹⁰https://github.com/TMContent/Lib_Vulnerabilities/commit/2e64495adc41ad74a517ddeb010d0368dd26f385

¹¹https://github.com/tm-sme/Lib_Vulnerabilities

Step 2) Create a local (forced) copy of the main repo and do the merge locally

- 6) git checkout -b upstreamVersion
- 7) git pull -f upstream master:upstreamVersion
- 8) git checkout mergeBranch
- 9) git merge upstreamVersion

which works:

```

1      Merge made by the 'recursive' strategy.
2      LICENSE.TXT | 50 ++++++=====
3      1 file changed, 50 insertions(+)
4      create mode 100644 LICENSE.TXT

```

Step 3) push the merged files to both repos

- 10) git push origin mergeBranch:mergeBranch

```

1      Counting objects: 7, done.
2      Delta compression using up to 4 threads.
3      Compressing objects: 100% (5/5), done.
4      Writing objects: 100% (5/5), 3.99 KiB | 0 bytes/s, done.
5      Total 5 (delta 2), reused 0 (delta 0)
6      To git@github.com:tm-sme/Lib_Vulnerabilities.git
7      * [new branch]      mergeBranch -> mergeBranch

```

- 11) push upstream mergeBranch:mergeBranch

```

1      Counting objects: 2971, done.
2      Delta compression using up to 4 threads.
3      Compressing objects: 100% (774/774), done.
4      Writing objects: 100% (2890/2890), 431.68 KiB | 0 bytes/s, done.
5      Total 2890 (delta 2199), reused 2801 (delta 2116)
6      To git@github.com:TMContent/Lib_Vulnerabilities.git
7      * [new branch]      mergeBranch -> mergeBranch

```

Step 4) merge into main branch of main repo

- 12) ... the next step was done on GitHub using a Pull Request on the [TMContent/Lib_Vulnerabilities](https://github.com/TMContent/Lib_Vulnerabilities)¹² repo

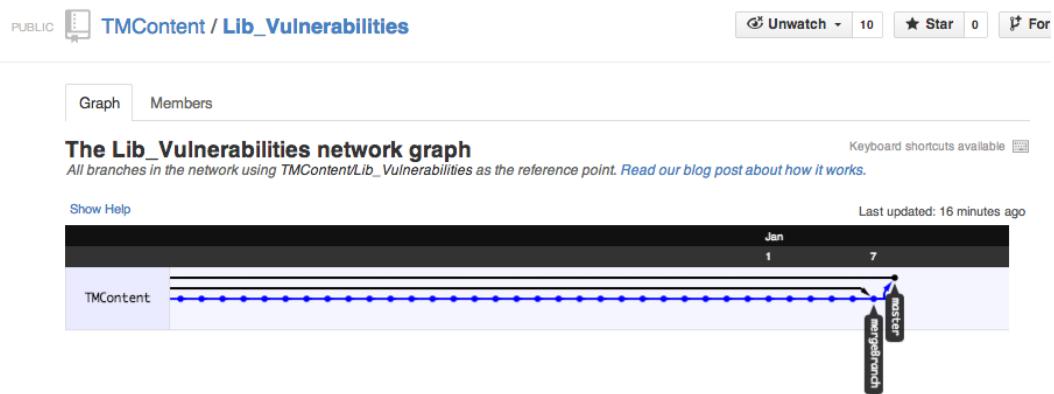
¹²https://github.com/TMContent/Lib_Vulnerabilities

```
1 git push upstream mergeBranch:master
```

Step 5) update local master and forked repo master

- 13) git checkout master
- 14) git pull upstream master:master
- 15) git push origin master:master

Here is what the main repo looks after the merge

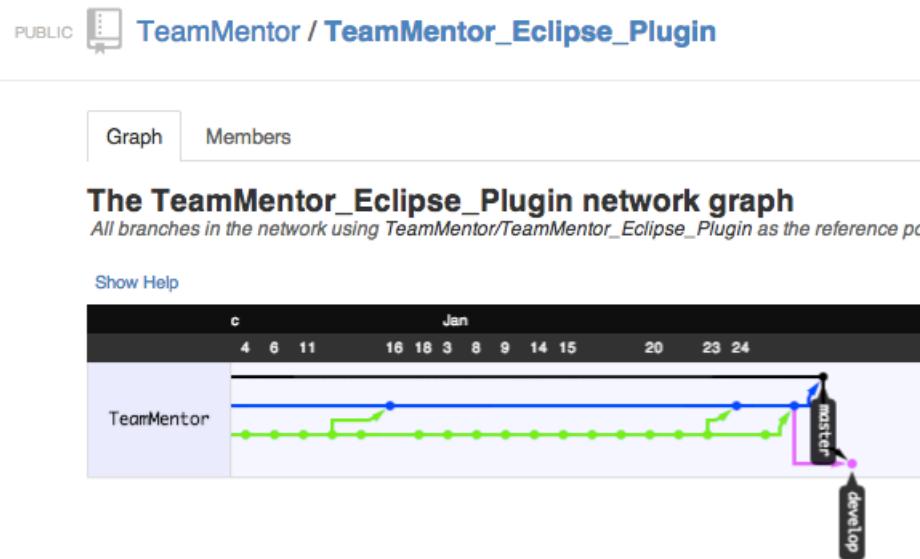


Updating the GitHub repos for the 1.6.0 release of the Eclipse Fortify Plugin

As you can see by the [recent eclipse related posts¹³](#), I have been working on a Plugin for Eclipse that shows [TeamMentor¹⁴](#) guidance to users that have access to the Fortify Eclipse plugin (and *.fpr files). We are now in the final stages of releasing the first public version (1.6.0) which is actually made of two parts: An [Eclipse Plugin builder¹⁵](#) (which is Open Source) and a small ‘Fortify Specific’ code-mapping script. Very soon these will be in separate projects, but for now they are all hosted at the [TeamMentor/TeamMentor_Eclipse_Plugin¹⁶](#).

This post is just to document the current GitHub development model and where to find the main parts of this release.

As mentioned above, the master version of the code is at [TeamMentor/TeamMentor_Eclipse_Plugin¹⁷](#) which currently looks like this:



One interesting point here is that for this release I did not use my main GitHub *DinisCruz*

¹³<http://blog.diniscruz.com/search?q=+eclipse>

¹⁴<https://teammentor.net/>

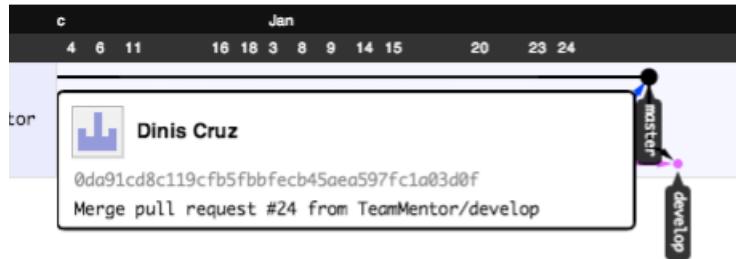
¹⁵<http://blog.diniscruz.com/2013/11/si-open-sources-eclipse-plugin.html>

¹⁶https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin

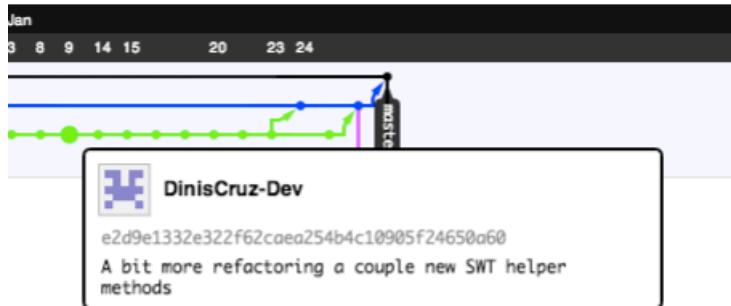
¹⁷https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin

account, but used instead a much less powerful GitHub *DinisCruz-Dev* account.

To see this in action, note how the Pull Request commits (into the master and develop branch) are made using the *DinisCruz* account:



... and the development commits are made using the *DinisCruz-Dev* account:



What I did was to fork into the *DinisCruz-Dev* account, the [TeamMentor/TeamMentor_Eclipse_Plugin¹⁸](https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin) repo:

¹⁸https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin

Graph Members

Members of the TeamMentor_Eclipse_Plugin Network

TeamMentor created TeamMentor_Eclipse_Plugin and everyone else forked it. TI

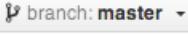
- TeamMentor / TeamMentor_Eclipse_Plugin
- DinisCruz-Dev / TeamMentor_Eclipse_Plugin

... which was then used during development (which in practice means that the _DinisCruz-Dev_ account does NOT have commit privileges to the release version of the code base)

PUBLIC  [DinisCruz-Dev / TeamMentor_Eclipse_Plugin](#)
forked from TeamMentor/TeamMentor_Eclipse_Plugin

Workspace files for TeamMentor Eclipse Plugin

 29 commits  10 branches  0 releases

  [TeamMentor_Eclipse_Plugin](#) / 

In terms of the 1.6.0 release, I also added a Git Tag to it (now possible to do via the GitHub web UI), so that this version can be easily accessed and downloaded from the [repo's Releases page](#)¹⁹:

¹⁹https://www.blogger.com/the%20https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin/releases

PUBLIC  TeamMentor / TeamMentor_Eclipse_Plugin Unwatch ▾ 6

Releases / Tags

Latest release

 1.6.0
-o 0da91cd

1.6.0 Release

 **DinisCruz** released this 23 minutes ago · **0 commits** to master since this release
Updating master with develop branch (1.6.0)

[Source code \(zip\)](#) [Source code \(tar.gz\)](#)

Pre-release

 v1.3.5
-o 0122dec

1.3.5 Release

 **DinisCruz-Dev** released this 3 months ago · **54 commits** to master since this release
First Beta release of the TeamMentor Eclipse Plugin

[Source code \(zip\)](#) [Source code \(tar.gz\)](#)

In order to help installation and deployment of this plugin in Eclipse, there is also this Eclipse Update site repo [TeamMentor/TeamMentor_Eclipse_Plugin_Deploy](#)²⁰

²⁰https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin_Deploy

[TeamMentor / TeamMentor_Eclipse_Plugin_Deploy](#) Unwatch ▾ 6

Deployment for install/update of TeamMentor Eclipse Plugin — Edit

56 commits 1 branch 1 release 3 contributors

branch: master [TeamMentor_Eclipse_Plugin_Deploy / +](#)

Version 1.6.0		
	DinisCruz-Dev authored 17 hours ago	latest commit 7eeadd6687 ↗
	Version 1.6.0	17 hours ago
	Version 1.6.0	17 hours ago
	Adding first version of 'Eclipse Plugin Update Site' files	4 months ago
	Version 1.6.0	17 hours ago
	Adding first version of 'Eclipse Plugin Update Site' files	4 months ago
	Version 1.6.0	17 hours ago
	Version 1.6.0	17 hours ago
	minor changes	6 days ago
	Version 1.6.0	17 hours ago

... which also contains the v.1.6.0 release tag:

[TeamMentor / TeamMentor_Eclipse_Plugin_Deploy](#) Unwatch ▾ 6

[Releases / Tags](#)

Latest release	
	1.6.0
↳	7eeadd6
1.6.0 Release	
DinisCruz released this 17 hours ago · 0 commits to master since this release	
	Source code (zip)
	Source code (tar.gz)

... and can be used inside eclipse using a local clone of this repo, or via this temp update site <https://eclipse-plugin-builder.scm.azurewebsites.net>²¹ (see more detailed installation instruc-

²¹<https://eclipse-plugin-builder.scm.azurewebsites.net>/

tions at:[TeamMentor Plugin and Builder v1.5.6 \(Source Code and Eclipse Update site\)](#)²²).

Now that this release is out of the way, I will try to write a number of blog posts that show how it works and how powerful the Eclipse Plugin Builder is (for example to add support for more tools or easily create eclipse plugins to help developers to write better/securer code)

²²<http://blog.diniscruz.com/2013/11/teammendor-plugin-and-builder-v156.html>

Updating GitHub Forks with latest commits from GitHub's 'parent' repo

One of the areas that tend to cause some problems with GitHub 'Forking model' workflow, is the need to have the *Forks* updated with the commits that have been added to the *Parent* repo (i.e. the repo that was used to create the *Fork* from).

To see real-word examples (and pains) of this issue, take a look at these posts:

- How to update a forked GitHub repo (in this case [tm-sme/Lib_Vulnerabilities](#))²³
- Syncing all releases to the same commit and Tag (for TeamMentor v3.4)²⁴
- Fixing the Merge conflict caused by one extra commit on TeamMentor master²⁵

For the example show below, I'm going to update the [DinisCruz-Dev/TeamMentor_Eclipse_Plugin](#)²⁶ repo which is a *Fork* of [TeamMentor/TeamMentor_Eclipse_Plugin](#)²⁷ (here also referenced as the *Parent* repo).

At the moment these repos are the stage shown in [Updating the GitHub repos for the 1.6.0 release of the Eclipse Fortify Plugin](#)²⁸ :

- The commits made on the *Fork* ([DinisCruz-Dev/TeamMentor_Eclipse_Plugin](#)²⁹) have been pushed into the *Parent* ([TeamMentor/TeamMentor_Eclipse_Plugin](#)³⁰) as a Pull Requests
- All Pull Requests have been merged back into the *Parent* repo (with even a [v1.6.0 tag created to mark the release](#)³¹)
- The *Fork* repo has NOT been updated with the final commits.

Here is what this looks like form the *Parent*'s repo [TeamMentor/TeamMentor_Eclipse_Plugin](#)³² point of view:

²³<http://blog.diniscruz.com/2014/01/how-to-update-forked-github-repo-in.html>

²⁴<http://blog.diniscruz.com/2013/10/syncing-all-releases-to-same-commit-and.html>

²⁵<http://blog.diniscruz.com/2013/10/fixing-merge-conflict-caused-by-one.html>

²⁶https://github.com/DinisCruz-Dev/TeamMentor_Eclipse_Plugin

²⁷https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin

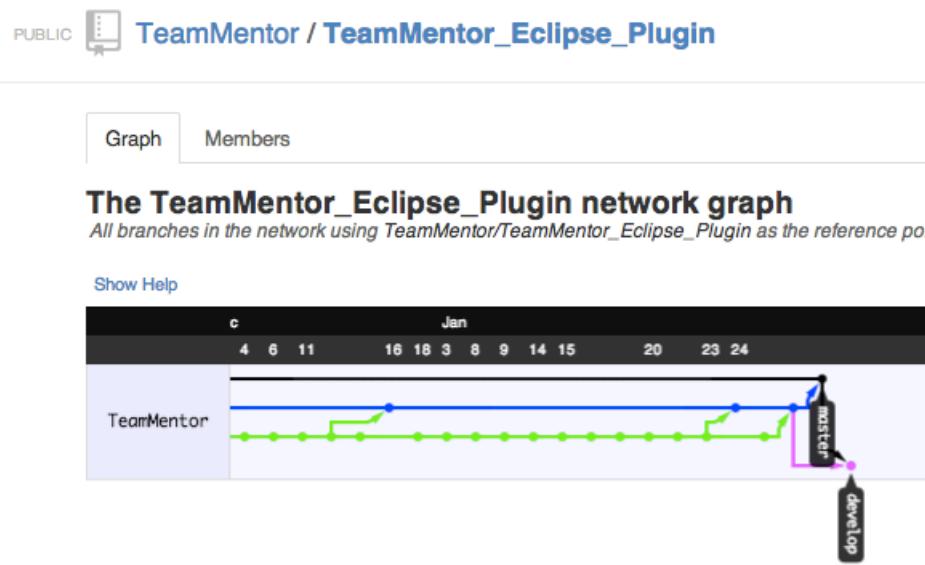
²⁸<http://blog.diniscruz.com/2014/01/updating-github-repos-for-160-release.html>

²⁹https://github.com/DinisCruz-Dev/TeamMentor_Eclipse_Plugin

³⁰https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin

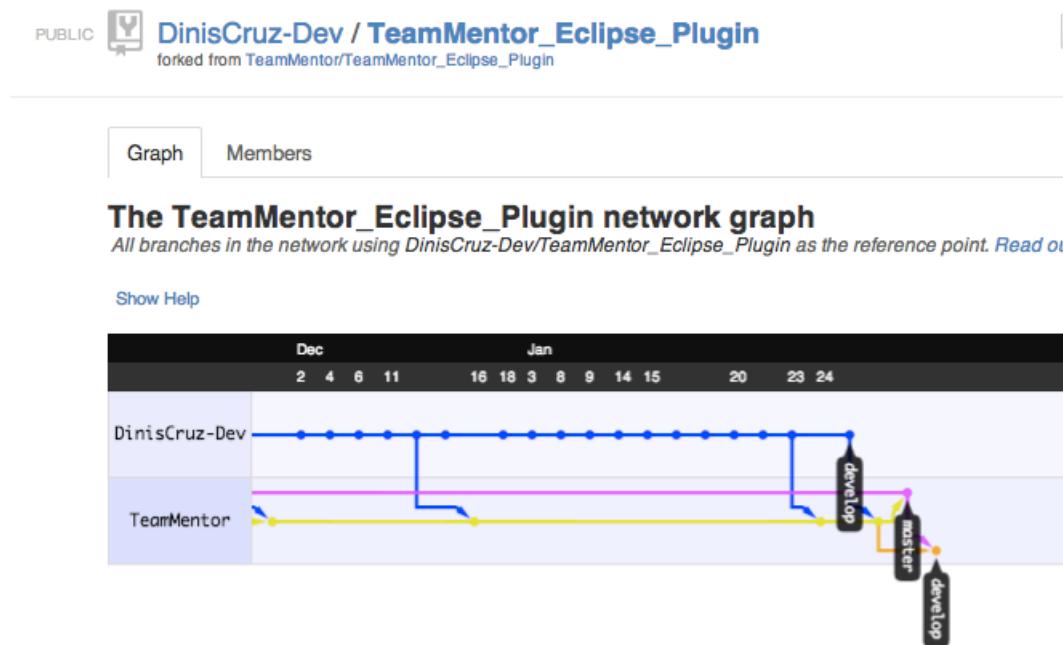
³¹https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin/releases/tag/1.6.0

³²https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin



And here is what this looks like from the *Forked* repo [DinisCruz-Dev/TeamMentor_Eclipse_Plugin](https://github.com/DinisCruz-Dev/TeamMentor_Eclipse_Plugin)³³ point of view:

³³https://github.com/DinisCruz-Dev/TeamMentor_Eclipse_Plugin



Looking at the common commits, might help to visualize what is going on.

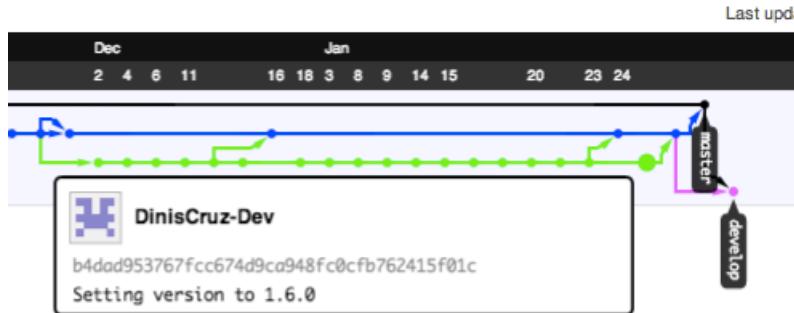
The last commit at the *Fork* is the [b4dad953767fcc674d9ca948fc0cfb762415f01c³⁴](#), which can be seen below as represented by the large LAST BLUE dot in the [DinisCruz-Dev/TeamMentor_Eclipse_Plugin³⁵](#) repo Network graph



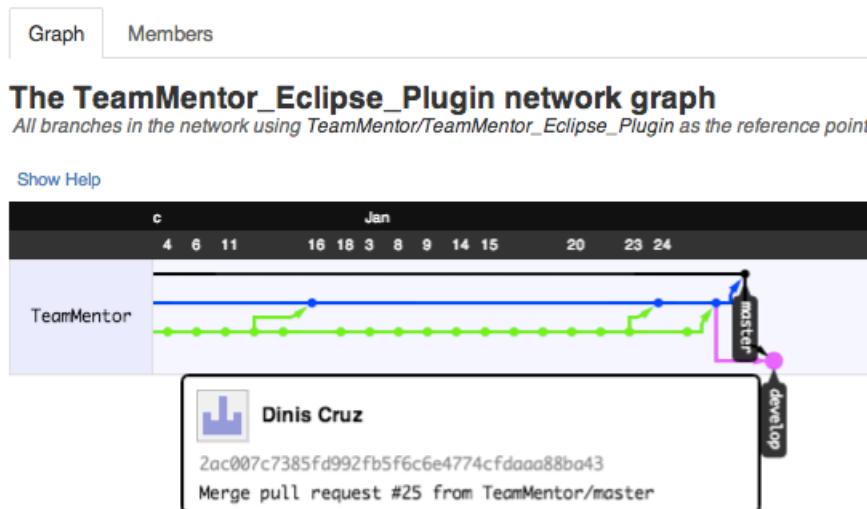
³⁴https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin/commit/b4dad953767fcc674d9ca948fc0cfb762415f01c

³⁵https://github.com/DinisCruz-Dev/TeamMentor_Eclipse_Plugin

But the same commit can be seen below as the large GREEN dot in the [TeamMentor/TeamMentor_Eclipse_Plugin](#)³⁶ *Parent* repo (note that it is not the last one in this repo):



At the moment the *Parent* repo is currently at the [2ac007c7385fd992fb5f6c6e4774cfdaaa88ba43](#)³⁷ commit (which doesn't exist in the *Forked* repo)



The practical consequences of this situation, is that the *Fork* is currently in an ‘incompatible’ state with its *Parent*, and it will not be possible to send Pull Requests/code-fixes upstream (note that this is ‘by design’ since Git does not allow merges when there are no common parents).

³⁶https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin

³⁷https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin/commit/2ac007c7385fd992fb5f6c6e4774cfdaaa88ba43

The solution is to do a Pull Request from '*Parent* to *Fork*' (ie. from [TeamMentor/TeamMentor_Eclipse_Plugin](#)³⁸ to [DinisCruz-Dev/TeamMentor_Eclipse_Plugin](#)³⁹), as seen below:

PUBLIC TeamMentor / TeamMentor_Eclipse_Plugin Unwatch ▾ 6 Star 0 Fork 1

base fork: [DinisCruz-Dev/TeamMentor_Eclipse_Plugin](#) ▾ base: [develop](#) ... head fork: [TeamMentor/TeamMentor_Eclipse_Plugin](#) ▾ compare: [develop](#)

Click to create a pull request for this comparison

8 commits 1 file changed 0 comments 2 contributors

In this case, there is only one update (made directly on the *Parent* repo) that needs to be merged into the *Forked* repo, and more importantly we get the desired [2ac007c7385fd992fb5f6c6e4774cfdaaa](#) commit:

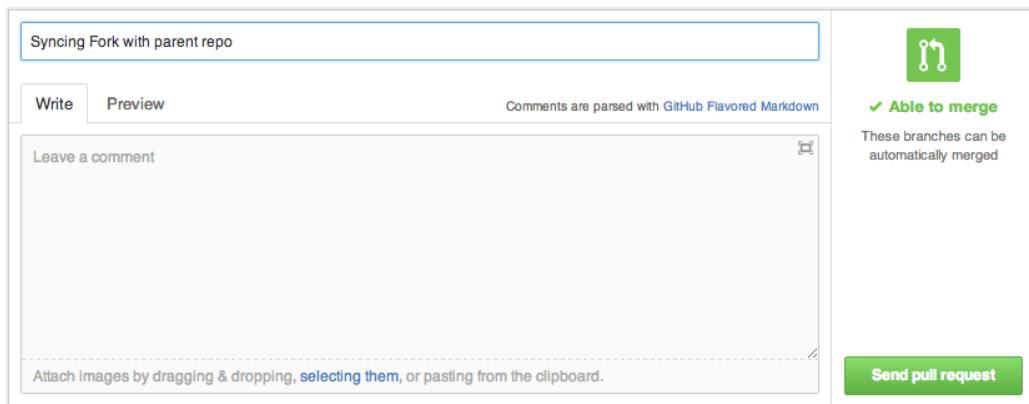
Jan 24, 2014			
	Merge pull request #22 from DinisCruz-Dev/develop	...	d2a0c95
	Merge pull request #23 from DinisCruz-Dev/develop	...	709681c
	Merge pull request #24 from TeamMentor/develop	...	0da91cd
	Merge pull request #25 from TeamMentor/master	...	2ac007c

Here is how I created the Pull Request:

³⁸https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin

³⁹https://github.com/DinisCruz-Dev/TeamMentor_Eclipse_Plugin

⁴⁰https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin/commit/2ac007c7385fd992fb5f6c6e4774cfdaaa88ba43



Which I then opened up in a browser logged in as *DinisCruz-Dev* (the *DinisCruz* account doesn't have GitHub privs to make this merge)

This repository Search or type a command Explore Gist Blog Help DinisCruz-Dev + - X ↗

DinisCruz-Dev / TeamMentor_Eclipse_Plugin forked from TeamMentor/TeamMentor_Eclipse_Plugin Unwatch 1 Star 0 Fork 1

Open DinisCruz wants to merge 8 commits into DinisCruz-Dev:develop from TeamMentor:develop #1 4 #1

Conversation 8 Commits 8 Files Changed 1

DinisCruz opened this pull request just now Syncing Fork with parent repo Edit
+ 4 additions - 0 deletions

No one is assigned No description given.

And since the commits are all compatible, I can just click on the 'Merge pull request' button:

Add more commits by pushing to the [develop](#) branch on [TeamMentor/TeamMentor_Eclipse_Plugin](#).



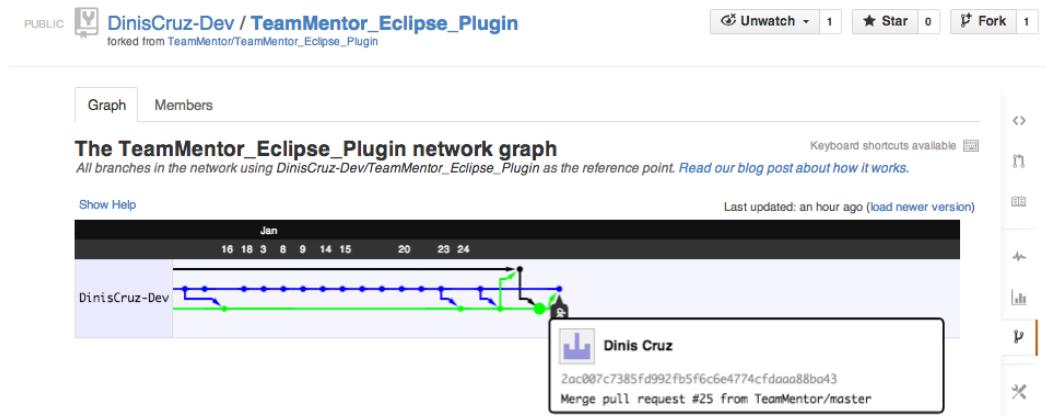
... to successfully apply the merge:

Merged DinisCruz-Dev merged commit `fe4356f` into `DinisCruz-Dev:develop` from `TeamMentor:develop` just now

Closed DinisCruz-Dev closed the pull request just now

Pull request successfully merged and closed
You're all set—the `TeamMentor:develop` branch can be safely deleted.

And now the Forked repo contains all commits that exist in the ‘parent’ repo (note the the `2ac007c7385fd992fb5f6c6e4774cfdaaa88ba43`⁴¹ commit below)



A final good house cleaning step is to also update the Master branch of the Fork:

⁴¹https://github.com/TeamMentor/TeamMentor_Eclipse_Plugin/commit/2ac007c7385fd992fb5f6c6e4774cfdaaa88ba43

PUBLIC DinisCruz-Dev / **TeamMentor_Eclipse_Plugin**
forked from TeamMentor/TeamMentor_Eclipse_Plugin

Unwatch 1 Star 0 Fork 1

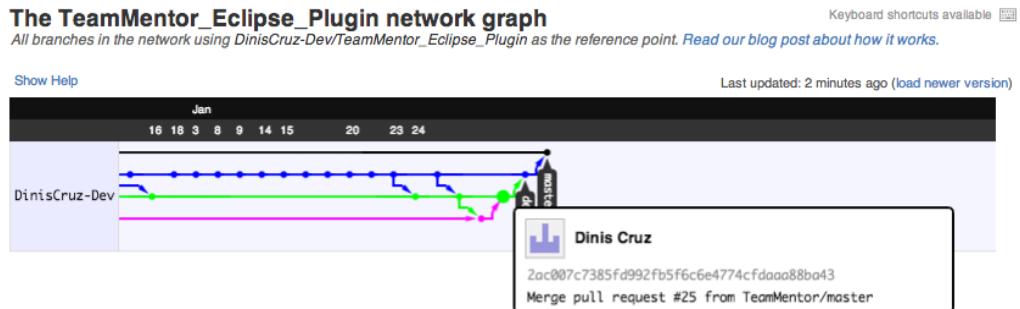
master ... develop Edit

Click to create a pull request for this comparison

56 commits 227 files changed 0 comments 2 contributors

Commits Files Changed Commit Comments

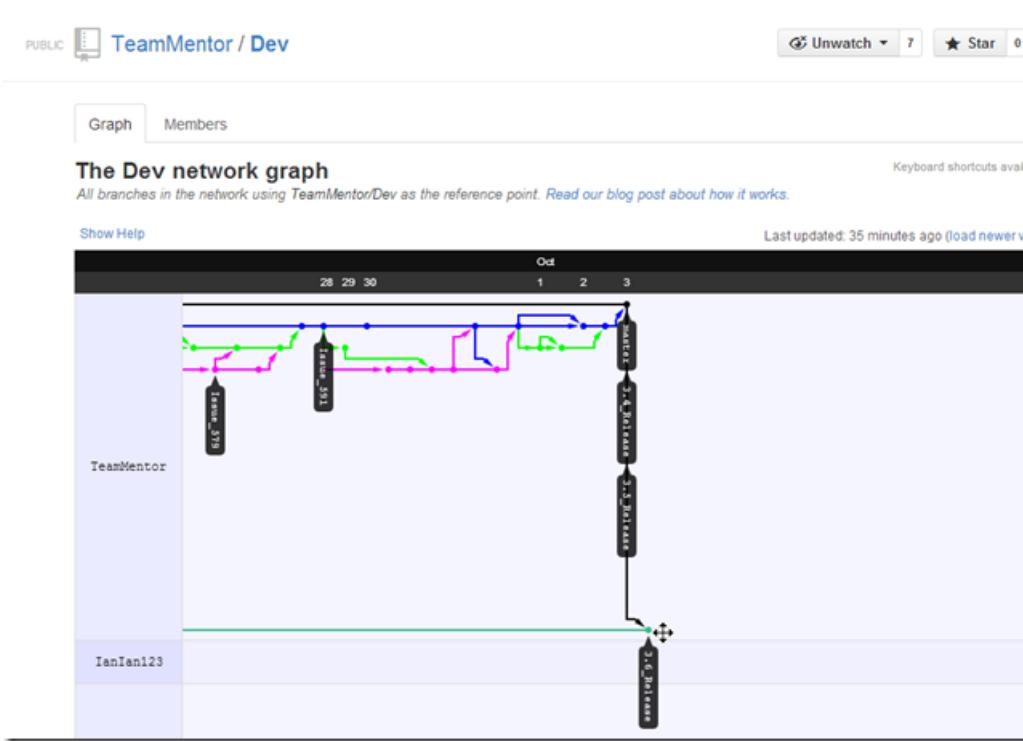
... which makes the final version of the graph look like this:



Note that all these steps could have been done using the git.exe command line, and in some cases, that is better, since we have more control over the creation of new commits on merge (for example note how every-time I merged a Pull Request in GitHub, a new commit was created! ... which is something that is not needed all the time)

For example I prefer when we can align the branches that are synced with the same commit, like what was done on [Syncing all releases to the same commit and Tag \(for TeamMentor v3.4\)](#)⁴² and shown below:

⁴²<http://blog.diniscruz.com/2013/10 syncing-all-releases-to-same-commit-and.html>



Using TeamMentor 3.4 TBot admin pages to load and sync a Library hosted on GitHub

Serge asked me to help making some changes to the [TeamMentor⁴³](#)'s Asp.NET 3.5 library, and since we need a test server to look at what might be changed (and run some scripts) this is a good time to show about how to use the TeamMentor's 3.4 Tbot pages to load a Library hosted on GitHub

I will also show, how once the TM server is configured with a library using a Git url, changes can be auto committed/pushed to that Git server, every-time there is a content edition using TM's web editors.

Step 1: Preparing the target TM server

Lets start with an Azure hosted TeamMentor server, for example this one:

The screenshot shows the TeamMentor web application running on an Azure website. The interface includes a navigation bar with links for Edit Mode, Change Password, Control Panel, and Logout. It also shows the TM 3.4 version and a user logged in as dinis.

Applied Filters: Shows filters for Technology (Any, .NET 2.0, ADO.NET 2.0, ASP.NET 2.0, Any, SQL Server 2000, Web Application) and Phase (Deployment, Design, Implementation, Test).

Guidance Libraries: A tree view listing various security topics under .NET 2.0, .NET 3.5, .NET 4.0, Android, C++, CWE, HTML5, iOS, Java, PCI DSS Compliance, PHP, Scala, and Vulnerabilities.

Search: A search bar with a magnifying glass icon.

Selected Article: A detailed view of an article titled "A Centralized Log Server Is Deployed".

Title	Technology	Phase
A Centralized Log Server Is Deployed	Any	Deployment
A Certificate Is Installed on the Database Server to Support SSL Communication	SQL Server 2000	Deployment
A Control Flow Analysis Is Performed	Any	Implementation
A Custom ASP.NET Policy Is Used to Access Non-SQL Server Databases	ADO.NET	Implementation

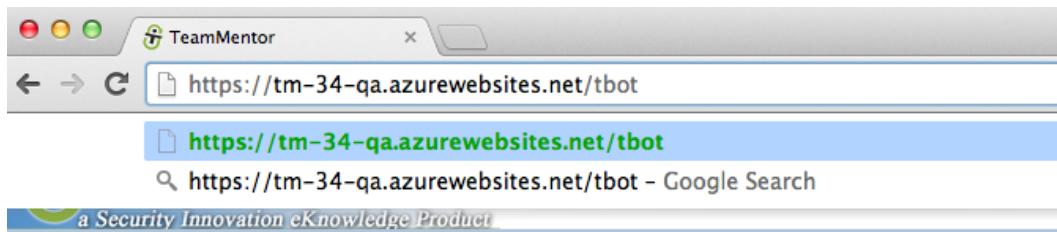
Selected Article Details:

- Applies to:**
 - PCI DSS Requirement 10.5
 - PCI DSS Requirement 10.5.1
 - PCI DSS Requirement 10.5.2
 - PCI DSS Requirement 10.5.3
 - PCI DSS Requirement 10.5.4
 - PCI DSS Requirement 10.6
 - PCI DSS Requirement 10.6.a
 - PCI DSS Requirement 10.6.b
 - PCI DSS Requirement 10.7
 - PCI DSS Requirement 10.7.a
 - PCI DSS Requirement 10.7.b
- What to Check For:** A centralized log server should be deployed.
- Why:** It may be possible to satisfy PCI DSS requirements.

Since we don't need all those libraries in there (and in fact we want to make changes to the .NET 3.5 Library), lets remove them all.

The easiest way to do it is to change the backend location of the TeamMentor XML files, which can easily be done by going into TBot:

⁴³<https://teammentor.net/>



... changing the *TMConfig.config* file *XmlLibrariesPath* value:

TBot - your friendly TeamMentor Bot

Editing TMConfig.Config file (and preview)

Important Note: This will edit the TMConfig.Config file located in the WebRoot (which if you have set the GitUserDataFolder used). Please contact support if you need help.

TMConfig.Config path : C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\app_data\TMConfig.Config

```
{
  "_type": "TeamMentor.CoreLib.TMConfig",
  "TMSetup": {
    "UseAppDataFolder": false,
    "TMLibraryDataVirtualPath": "..\\..",
    "XmlLibrariesPath": "TM_Libraries_Test", // was "TM_Libraries"
    "UserDataPath": "User_Data",
    "LibrariesUploadedFiles": "LibrariesUploadedFiles",
    "EnableGZipForWebServices": true,
    "Enable304Redirects": true
  }
}
```

Save data loaded

... and now, after the cache is reloaded:

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, Including User data) Done: "In the library 'C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase' there are 0 library(ies), 0 views and 0 GuidanceItems"

... there will be no articles on this instance of TM:

TIP: If you want to quickly add a test Library to TM, you can use the old `/admin` panel option to install Libraries from a link or zip. Here is how I quickly installed the OWASP Library, by clicking on the OWASP link in this page:

User Management
 - [My Account](#)
 - [Manage Users](#)
 - [Create Multiple Users](#)

BackEnd
 - [Set Library Path](#)
 - [Set UserData Path](#)
 - [Admin Tasks](#)

Developer Utils
 - [Install/Upload Libraries](#)

- [Firebug Lite](#)
- [Web Editor](#)
- [C# REPL](#)
- [GitHub Sync](#)
- [TeamMentor QUnit Tests](#)
- [UIAutomation \(Admin\)](#)
- [UIAutomation \(User\)](#)

Misc links:

Install Library

Library Zip files uploaded to this server (click to install)

Current Libraries Zip Folder(server side): C:\DWASFiles\Sites\tm-34-qal\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\Xml\Database\LibrariesUpload

Note that instalation from Zip will override existing libraries with the same name. [refres](#)

- [SI Library - 3.4.zip](#)

Install Library from File or Url

URLs for Libraries to install: [OWASP Top 20 Vulnerabilities](#) [TM Documentation](#)

File (server-side) or Url: https://github.com/TeamMentor/OWASP_Library/zipball

Unzip Password:

> Library installed was successful

... which makes my test TM instance now look like this:

Selected Article

A Secure Key Storage Location Is Used

What to Check For

Ensure that application keys are stored in a well defined location, such as the encrypted sections of the application's `web.config` or encrypted in the Windows Registry. Verify that they are not hard coded into the application or stored as plaintext inside configuration files.

Why

When faced with encryption, intruders attack the most vulnerable aspect: key management. Because keys are used to encrypt sensitive data, the keys themselves become sensitive data that need to be

Step 2. Create a Fork of the Target Library

In this case the TM Library I want to make changes to is the private repo [tm-sme/Lib_.NET_-3.5](https://github.com/tm-sme/Lib_.NET_3.5) which is the one used by SI's SME team to make changes into the next version of TM (the idea is that I will make changes in my Fork which I will then issue a Pull Request to this version)

Code

Pull Requests 0

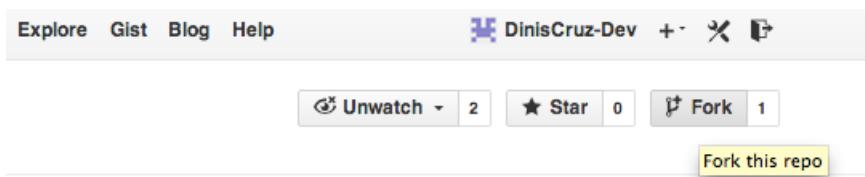
Wiki

Pulse

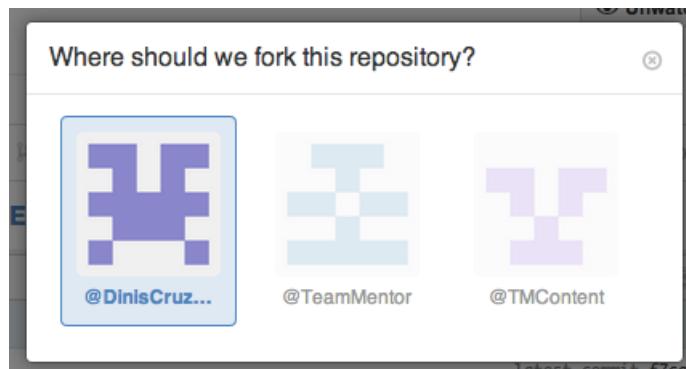
Graphs

Network

Next, logged in as **DinisCruz-Dev** (which is my day-to-day GitHub not-very-privileged account), I clicked on the Fork link:



... chose the *DinisCruz-Dev* as the Fork target:



... and after a couple seconds I had a Fork of the [tm-sme/Lib.NET_3](https://github.com/tm-sme/Lib.NET_3) repo at [DinisCruz-Dev/Lib.NET_3.5](https://github.com/DinisCruz-Dev/Lib.NET_3.5)

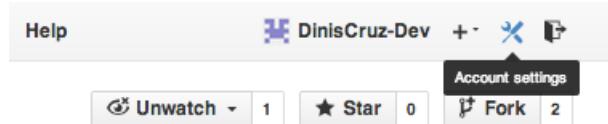
A screenshot of a GitHub repository page for 'DinisCruz-Dev / Lib_.NET_3.5'. The page shows basic repository details: 3 commits, 1 branch, 0 releases, and 1 contributor. The 'branch: master' dropdown is set to 'Lib_.NET_3.5'. The commit history shows a single commit from 'Attack' adding 'Added Library Files' a year ago. On the right side, there are links for 'Code', 'Pull Requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Network'.

Step 3: Configuring TeamMentor to load a Library from GitHub

This next step is a bit different if the IIS user account of the target server is configured to use SSH, but since Azure doesn't seem to support it, we will need to use HTTPS and hardcoded passwords to do this.

The good news is that GitHub now provides a nice way to create temp hard passwords, so I can use that on this blog post :)

On the GitHub's Account Settings for the *DinisCruz-Dev* user:



I clicked on the *Create new Token* button from the *Personal Access Tokens* section (part the *Applications* area)

A screenshot of the GitHub account settings page for 'DinisCruz-Dev'. The left sidebar has a 'Applications' section selected. The main content area shows the 'Personal Access Tokens' section, which contains a 'Create new token' button. Below it is a note about Personal Access Tokens functioning like OAuth tokens. The 'Authorized applications' section lists 'Windows Azure' with a 'Revoke' button. The 'GitHub applications' section lists 'Gist' with a 'Revoke' button.

... entered a name for it (the text in this page provides a good explanation for how this token should be used)

Create a new Personal Access Token

Personal Access Tokens function like ordinary OAuth access tokens. They can be used instead of a password for git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Enter a name for the token below so you can keep track of where you used it and revoke the token if necessary.

Token description:

Create Token

... and after it was created:

Developer applications

Do you want to develop an application that uses the [GitHub API?](#) [Register an application](#) to generate OAuth tokens.

[Register new application](#)

Personal Access Tokens

Personal Access Tokens function like ordinary OAuth access tokens. They can be used instead of a password for git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

124f9ce43f8cecd7f56b2a9e412118b01f72c...	Edit	Delete
--	----------------------	------------------------

... I copied it into the clipboard:

Personal Access Tokens

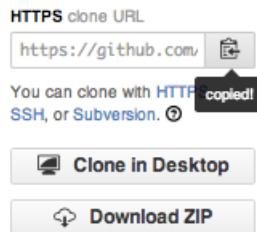
Personal Access Tokens function like ordinary OAuth access tokens. They can be used instead of a password for git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

124f9ce43f8cecd7f56b2a9e412118b01f72c...	Edit	Delete
--	----------------------	------------------------

copied!

In this case it the token is **124f9ce43f8cecd7f56b2a9e412118b01f72cef7**

Back in the [DinisCruz-Dev/Lib.NET_3.5](https://github.com/DinisCruz-Dev/Lib.NET_3.5) main page I clicked on *copy to clipboard* button that is close to the clone URL:



... which is this case is https://github.com/DinisCruz-Dev/Lib_.NET_3.5.git

Since we need to use this from the Azure instance, we need to hard-code the username and password using the syntax `https://[username]:[passwordOrToken]@github.com/DinisCruz-Dev/Lib_.NET_3.5.git`

... which in this case will be

`https://DinisCruz-Dev:124f9ce43f8cecd7f56b2a9e412118b01f72cef7@github.com/DinisCruz-Dev/Lib_.NET_3.5.git`

Next step is to go into TBot's *Edit SecretData* page:

The screenshot shows a web browser window with the URL `https://tm-34-qa.azurewebsites.net/rest/tbot/run/Edit_SecretData`. The page title is "TBot - your friendly TeamMentor Bot". The main content area is titled "Editing an Secret Data". It contains a JSON editor with the following data:

```
{
  "Libraries_Git.Repositories": [],
  "Rijndael_IV": "0I8KkgIuxrHESDK1rzTs8g==",
  "Rijndael_Key": "5OIRB3OZQWzrdQDJCCanjOkQXDUB4gHReQ6uBRD7a1g==",
  "SMTP_Password": null,
  "SMTP_Server": "smtp.sendgrid.net",
  "SMTP_UserName": "TeamMentor"
}
```

To the right of the JSON editor, there is a blue "Save" button and the text "data loaded".

... enter the Git url in the *Libraries_Git.Repositories* field, click Save:

TBot - your friendly TeamMentor Bot

Editing an Secret Data

```
{
  "Libraries_Git.Repositories": ["https://DinisCruz-
Dev:124f9ce43f8cecd7f56b2a9e412118b01f72cef7@github.com/DinisCruz-
Dev/Lib_.NET_3.5.git"],
  "Rijndael_IV": "0I8KkgluxrHESDK1rzTs8g==",
  "Rijndael_Key": "50IRB3OZQWzrdQDJCCaniOkQXDUB4gHReQ6uBRD7a1g=",
  "SMTP_Password": null,
  "SMTP_Server": "smtp.sendgrid.net",
  "SMTP_UserName": "TeamMentor"
}
```

Save data saved: true

... go into the *Reload Server Objects*, click on the Reload Cache button

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, including User data)

Reloading UserData

... which should take a little bit (depending on the size of the Library and server-side network connection speed)

.. until a message shows up that says how many libraries and GuidanceItems/Articles exist in the current server (which means that the git clone was successful and the TM server cache was reloaded)

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, including User data) Done: "In the library 'C:\\DWASFiles\\Sites\\tm-34-q\\VirtualDirectory0\\site\\wwwroot\\App_Data\\Library_Data\\XmlDatabase' there are 2 library(es), 64 views and 1954 Guidanceitems"

Opening up TeamMentor shows that we now have the new Library installed in this Azure server:

The screenshot shows the TeamMentor web application interface. On the left, there's a sidebar with 'Applied Filters' (Technology: Any, Phase: Deployment), 'Guidance Libraries' (including .NET 3.5, OWASP Top 10, PCI DSS Code Review, Security Engineering, Top 5 Rich Client Vulnerabilities, Top 5 Web Service Vulnerabilities, and OWASP), and a search bar. The main content area displays a list of articles. One article is selected: 'A Centralized Log Server Is Deployed'. The article details its application to PCI DSS Requirements 10.5 through 10.7.b. Below the article, under 'What to Check For', it says: 'A centralized log server should be deployed.'

Step 4: Configuring server to auto push commit changes into GitHub

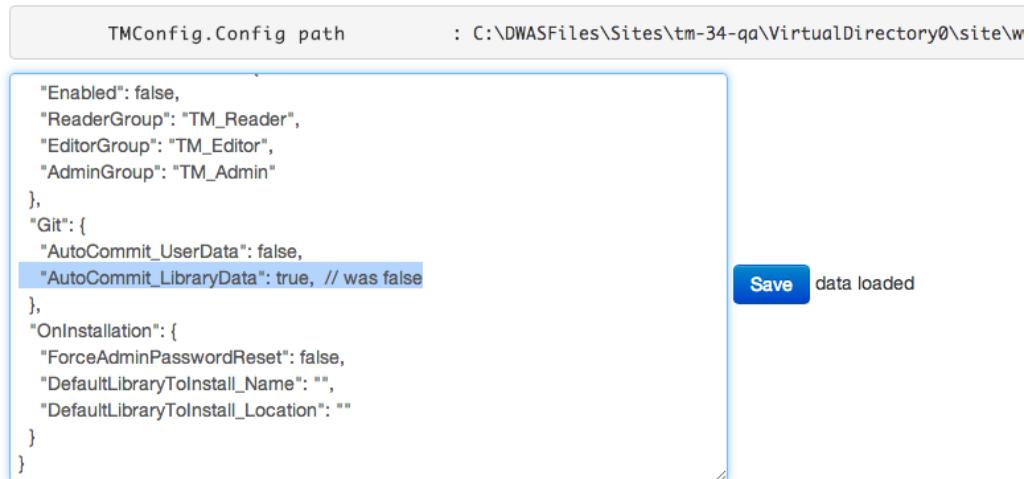
By default changes made on the server are not immediately pushed into the host server.

That behavior can be changed by setting to *true* the *AutoCommit_LibraryData* setting from the *TMConfig.config* file:

TBot - your friendly TeamMentor Bot

Editing TMConfig.Config file (and preview)

Important Note: This will edit the TMConfig.Config file **located in the WebRoot** (which if you have set the GitUserLocation used). Please contact support if you need help.



The screenshot shows a code editor interface for editing the TMConfig.Config file. The path is listed as 'TMConfig.Config path : C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\w'. The code editor displays the following JSON configuration:

```
"Enabled": false,
"ReaderGroup": "TM_Reader",
"EditorGroup": "TM_Editor",
"AdminGroup": "TM_Admin"
},
"Git": {
    "AutoCommit_UserData": false,
    "AutoCommit_LibraryData": true, // was false
},
"OnInstallation": {
    "ForceAdminPasswordReset": false,
    "DefaultLibraryToInstall_Name": "",
    "DefaultLibraryToInstall_Location": ""
}
```

A blue box highlights the line 'AutoCommit_LibraryData': true, // was false. To the right of the editor, there is a 'Save' button and the message 'data loaded'.

Once that is set (and after a server restart or TMConfig settings reload) changes made to TM articles will be auto-committed locally and pushed to GitHub as soon as possible (i.e. there is a bit of a delay in case there are multiple edits going on at the same time).

So see this in action, lets open an article:

The screenshot shows a web browser window with the URL <https://tm-34-qa.azurewebsites.net/article/00ae3320-ef51-441f-b186-ffd872f1c33b>. The page header features the TeamMentor logo and the text "a Security Innovation eKnowledge Product". On the right side, there are two buttons: "Edit WYSIWYG" and "Edit Source".

Do Not Divulge Exception Details to Clients in Production

Technology	Category	Phase	Type
WCF 3.5	Error and Exception Management	Implementation	Checklist Item

Applies to

- Microsoft® Windows Communication Foundation (WCF) 3.5
- Microsoft Visual Studio® 2008

What to Check For

Check to ensure that your application does not divulge exception details to clients in production.

... click on the *Edit WYSIWYG* link:

The screenshot shows the WYSIWYG editor interface for the article. At the top, it displays the URL <https://tm-34-qa.azurewebsites.net/editor/00ae3320-ef51-441f-b186-ffd872f1c33b>. The page header is identical to the previous screenshot. On the right, there are three buttons: "Edit Source", "View Article", and "Save Changes".

The main area contains a form with fields for "Title" (set to "Do Not Divulge Exception Details to Clients in Production"), "Technology" (set to "WCF 3.5"), "Phase" (set to "Implementation"), "Type" (set to "Checklist Item"), and "Category" (set to "Error and Exception Mana"). Below the form is a rich text editor toolbar with various formatting options like bold, italic, underline, and lists.

Applies to

- Microsoft® Windows Communication Foundation (WCF) 3.5
- Microsoft Visual Studio® 2008

What to Check For

Check to ensure that your application does not divulge exception details to clients in production.

... and make a change to the article (see '*THIS TEST*' below) and click on the *Save Changes* button

The screenshot shows the TeamMentor article editor interface. At the top, there's a logo for 'TeamMentor' and three buttons: 'Edit Source', 'View Article', and 'Save Changes'. Below the header is a table with columns for Title, Technology, Phase, Type, and Category. The Title is 'Do Not Divulge Exception Details to Clients in Production'. The Technology dropdown shows 'WCF 3.5'. The Phase dropdown shows 'Implementation'. The Type dropdown shows 'Checklist Item'. The Category dropdown shows 'Error and Exception Mana'. Below the table is a toolbar with various icons for editing.

Applies to (THIS TEST)

- Microsoft® Windows Communication Foundation (WCF) 3.5
- Microsoft Visual Studio® 2008

What to Check For

Check to ensure that your application does not divulge exception details to clients in production.

Once that is done, go back into the GitHub repo site, open the Commits page ([https://github.com/DinisCruz-Dev/Lib.NET_3.5/commits/master](https://github.com/DinisCruz-Dev/Lib.NET_3.5/commits/master)) and notice that there is an extra commit done *just now* (note: in TM 3.4 the commit is done under the server IIS user Git settings, which is usually not set, hence the value ‘unknown user’, in TM 3.6 there is already a fix to use the current TM Logged in user)

The screenshot shows the GitHub commits page for the 'Lib_.NET_3.5' repository. It lists several commits:

- Jan 29, 2014**: A commit by 'unknown-user' titled 'changed: Checklist Item/00ae3320-ef51-441f-b186-ffd872f1c33b.xml' with commit hash '605c645c4f'. It includes a 'Browse code' link.
- Jun 10, 2013**: A commit by 'unknown-user' titled 'changed: .NET 3.5.xml' with commit hash 'f7cad5a8d5'. It includes a 'Browse code' link.
- Nov 28, 2012**: A commit by 'Roman87' titled 'Added Library Files' with commit hash '37027a5ec5'. It includes a 'Browse code' link.

To really see the ‘real time’ commits and pushes, go back to the TM Article and make another change (this time around I’m using the ‘**NotePad**’ editor):

```

<h1>Applies to</h1>
<ul>
    <li>Microsoft® Windows Communication Foundation (WCF) 3.5</li>
</ul>
<ul>
    <li>Microsoft Visual Studio® 2008</li>
</ul>
<h1>What to Check For</h1>
<p>>Check to ensure that your application does not divulge exception details to clients in production.</p>
<hr />
<p>Adapted from Microsoft patterns & practices guidance.</p>

```

After a couple seconds from Saving the changes, another commit will exist in GitHub:

Date	Commit Message	Author	SHA	Action
Jan 29, 2014	changed: Checklist Item/00ae3320-ef51-441f-b186-ffd872f1c33b.xml unknown-user authored just now		3085370102	Browse code
Jan 29, 2014	changed: Checklist Item/00ae3320-ef51-441f-b186-ffd872f1c33b.xml unknown-user authored 2 minutes ago		605c645c4f	Browse code
Jun 10, 2013	changed: .NET 3.5.xml unknown-user authored 8 months ago		f7cad5a8d5	Browse code
Nov 28, 2012	Added Library Files Roman87 authored a year ago	Roman87	37027a5ec5	Browse code
Nov 14, 2012	Initial commit SergeTruth authored a year ago	SergeTruth	b4e180ef50	Browse code

... which is made of the user changes (i.e the diff of the changes made on the TM web interface)

The screenshot shows a GitHub pull request interface. At the top, it says "DinisCruz-Dev / Lib_.NET_3.5" and "forked from lm-sime/Lib_.NET_3.5". There are buttons for "Unwatch" (1), "Star" (0), and "Fork" (2). Below this, the pull request title is "changed: Checklist Item/00ae3320-ef51-441f-b186-ffd872f1c33b.xml" and it's authored by "unknown-user" just now. It has 1 parent, commit 30853701021a34f3405acfab9a3f31c5cee0f7f, and 2 deletions. A "Show Diff Stats" button is present. The main area shows a diff between Checklist Item/00ae3320-ef51-441f-b186-ffd872f1c33b.xml and master. The diff highlights changes in line 19 where content is sanitized from HTML to XML.

```

@@ -16,8 +16,8 @@
 16      <Status />
 17      <Source>SI</Source>
 18      </Metadata>
 19 -    <Content Sanitized="true" DataType="Html">
 20 -        <Data><![CDATA[<h1>Applies to (THIS TEST)</h1>
 21 +    <Content Sanitized="true" DataType="html">
 22 +        <Data><![CDATA[<h1>Applies to</h1>
 23             <ul>
 24                 <li>Microsoft® Windows Communication Foundation (WCF) 3.5
 25             </li>

```

Bonus Feature: Quick restore of an TeamMentor website

Not sure if it obvious by now, but what we have created here is a live version of TM whose content changes are being automatically synced (i.e backed up) into an external Git repository.

This means that if we completely lost the current website (let's say that Azure went down, or we had another episode of [AzureGate - how Azure's 'subscription upgrade' crazy mode caused us to stop using Azure for VM Hosting \(and Git+GitHub saved the day\)](#)⁴⁴), we could create a new instance of this TM website by just:

1. Creating a new Website using the latest release version from <https://github.com/TeamMentor/Master>
2. Logging in as Admin and in TBot configure add adding the GitRepo Url to the SecretData's *Libraries_Git.Repositories* value
3. Reload the cache or restart the server

That's it :)

And with Azure's APIs, this could all be scripted, which would make it even faster :)

⁴⁴<http://blog.diniscruz.com/2013/05/azuregate-how-azures-subscription.html>

Using TeamMentor 3.4 TBot admin pages to load and sync UserData with a GitHub hosted repo

Continuing from where [Using TeamMentor 3.4 TBot admin pages to load and sync a Library hosted on GitHub⁴⁵](#) left, this post shows how to use the same technique to sync TeamMentor's UserData with a GitHub repo.

For more details on how the *UserData repo/folder* fits within TeamMentor's architecture, see these posts:

- * [Writing RazorSharp script to import TeamMentor users⁴⁶](#) * [Creating QA versions of TeamMentor UserData repository, and using branches to show/test the multiple config options⁴⁷](#)
- * [Creating a version TeamMentor which uses the new GitUserData.config file⁴⁸](#) * [Practical Example of using Web CSharpREPL in TeamMentor's development/customizations⁴⁹](#) * [Using CSharpRepl to batch change TeamMentor's users email and settings⁵⁰](#) * [Running Customized C# code loaded from TeamMentor's UserData repository⁵¹](#) * [Using NGit to create native Git support in Azure deployed app \(with automatic pushes and pulls\)⁵²](#)

Step 1: Create UserData repo in GitHub

The first task is to create a **Private** repo to hold the *UserData* contents.

Important: Because it will contain sensitive data about the target TeamMentor instance (like password hashes, session IDs, emails, user activity tracking, SMTP account details and encryption key/salt), don't create a **Public** repo!

In GitHub, login into the desired account and go to the *New Repository* page:

⁴⁵<http://blog.diniscruz.com/2014/01/using-teammendor-34-tbot-admin-pages-to.html>

⁴⁶<http://blog.diniscruz.com/2013/03/writing-razorsharp-script-to-import.html>

⁴⁷<http://blog.diniscruz.com/2013/04/creating-qa-versions-of-teammendor.html>

⁴⁸<http://blog.diniscruz.com/2013/03/creating-version-teammendor-which-uses.html>

⁴⁹<http://blog.diniscruz.com/2013/04/practical-example-of-using-web.html>

⁵⁰<http://blog.diniscruz.com/2013/04/using-csharpapl-to-batch-change.html>

⁵¹<http://blog.diniscruz.com/2013/04/running-customized-c-code-loaded-from.html>

⁵²<http://blog.diniscruz.com/2013/03/using-ngit-to-create-native-git-support.html>

The screenshot shows the GitHub repository creation process. The URL in the address bar is <https://github.com/new>. The repository name is "Site_TM_34_QA_Azure". The owner is "DinisCruz-Dev". The repository is set to "PRIVATE". The description is "User Data repo for https://tm-34-qa.azurewebsites.net website". The visibility is set to "Private".

In the repository details view, there is a "Quick setup" section with instructions to add a README file. It also shows the repository URL: https://github.com/DinisCruz-Dev/Site_TM_34_QA_Azure. Below this, there is a "Create a new repository on the command line" section with a code block:

```
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/DinisCruz-Dev/Site_TM_34_QA_Azure.git
git push -u origin master
```

There is also a "Push an existing repository from the command line" section with a code block:

```
git remote add origin https://github.com/DinisCruz-Dev/Site_TM_34_QA_Azure.git
git push -u origin master
```

On the right side, there are links to "Code", "Issues", "Pull Requests", "Wiki", "Pulse", "Graphs", "Network", and "Settings".

Step 2: Make sure the repo is created with at least one file

If you chose the option to add a default README file in the previous step, you can ignore this, but if you didn't you will need to make sure that this repo has at least one branch and one file (or the Git Clone from TeamMentor will be left in a non-working state).

The good news is that you can easily do that from GitHub's interface.

On the **Quick setup** section, click on the **README me** link:

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTP SSH https://github.com/DinisCruz-Dev/Site_TM_34_QA. [Edit](#)

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

... which will open a web UI where the README.md file can be created:

PRIVATE [Unwatch](#) 1 ★ Star 0

DinisCruz-Dev / Site_TM_34_QA_Azure

Site_TM_34_QA_Azure / README.md or cancel

Spaces 2 No wrap

```
1 Site_TM_34_QA_Azure
2 -----
3 User Data repo for https://tm-34-qa.azurewebsites.net website
4
5
```

And after clicking on **Commit New File**

Commit summary:
Create README.md

Extended description: (optional)

DinisCruz-Dev dinis.cruz+dev@owasp.org

Cancel Commit New File

... the target repo is now in state that can be used by TeamMentor

User Data repo for <https://tm-34-qa.azurewebsites.net> website — Edit

1 commit 1 branch 0 releases 1 contributor

branch: master → [Site_TM_34_QA_Azure](#)

Create README.md

DinisCruz-Dev authored just now latest commit 4b65666159

[README.md](#) Create README.md just now

[README.md](#)

Site_TM_34_QA_Azure

User Data repo for <https://tm-34-qa.azurewebsites.net> website

Step 3: Create a GitHub Personal Access Token to be used to access this account from TM server

As with the [previous scenario⁵³](#), that is done on this Admin page

Create a new Personal Access Token

Personal Access Tokens function like ordinary OAuth access tokens. They can be used instead of a password for git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Enter a name for the token below so you can keep track of where you used it and revoke the token if necessary.

Token description:

Temp Token for Blog Post

Create Token

On the resulting page, copy the token (in this case c78fa4d5dcf1b9f521a99396d667a00297734a2b)

⁵³<http://blog.diniscruz.com/2014/01/using-teammendor-34-tbot-admin-pages-to.html>

The screenshot shows the GitHub 'Personal Access Tokens' page. At the top right is a 'Create new token' button. Below it is a text area explaining what Personal Access Tokens are. A token entry is shown with the ID 'c78fa4d5dcf1b9f521a99396d667a00297734...', a label 'Temp Token for Blog Post – Edit', and a 'Delete' button. A 'Copy Token' button is also present.

This Token will be used together with the HTTPS git url (https://github.com/DinisCruz-Dev/Site_TM_34_QA_Azure.git⁵⁴) in the format: http://{username}:{password/token}@github.com/{GitRepo}. In this case:

https://DinisCruz-Dev:c78fa4d5dcf1b9f521a99396d667a00297734a2b@github.com/DinisCruz-Dev/Site_TM_34_QA_Azure.git

Step 4: Configure TeamMentor Server to use GitHub's UserData repo:

next open TBot's *Edit GitUserLocation* page:

The screenshot shows the TBot control center. The URL in the browser bar is <https://tm-34-qa.azurewebsites.net/rest/tbot/run/Commands>. The main area displays the title 'TBot - your friendly TeamMentor Bot' and a section titled 'Available TBot Commands'. It lists several commands in a grid:

Admin	Import Legacy Users	Reload Server Objects
Admin\Config	Edit GitUserLocation	Edit SecretData
Admin\LiveData	ActiveSessions	DebugInfo

... and enter the Git Url (shown above) in the *Git User Location* textbox:

⁵⁴https://github.com/DinisCruz-Dev/Site_TM_34_QA_Azure.git

TBot - your friendly TeamMentor Bot

Editing an Secret Data

Git User Location:

`https://DinisCruz-Dev: c78fa4d5dcf1b9f521a99396d667a00297734a2b@github.com`

Save data loaded

Important: In order to keep the UserData up-to-date, it is also needed to set the *TmConfig.config*'s *AutoCommit_UserData* value to **true**

Editing TMConfig.Config file (and preview)

Important Note: This will edit the TMConfig.Config file located in the WebRoot (which if you have set the GitUserLocatior

TMConfig.Config path : C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\ww

```
},
"WindowsAuthentication": {
  "Enabled": false,
  "ReaderGroup": "TM_Reader",
  "EditorGroup": "TM_Editor",
  "AdminGroup": "TM_Admin"
},
"Git": {
  "AutoCommit_UserData": true,
  "AutoCommit_LibraryData": true
},
"OnInstallation": {
  "ForceAdminPasswordReset": false,
  "DefaultLibraryToInstall_Name": "",
  "DefaultLibraryToInstall_Location": ""
}
```

Save data loaded

Once that is done, reload the server cache (which will trigger the UserData setup):

TBot - your friendly TeamMentor Bot

Reloading Server objects

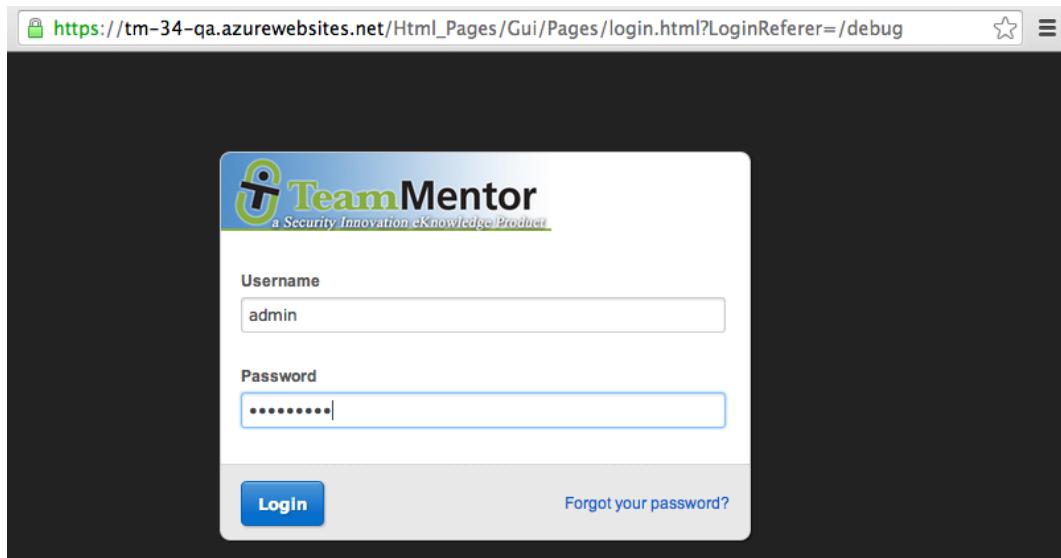
Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, including User data)

Reloading UserData

Once that is completed, you will notice that you are logged out from TM.



This happened because a new *UserData* user store was created which didn't have any accounts. In those cases TeamMentor server engine will create a default admin account using the details provided in *TMConfig.config*:

TBot - your friendly TeamMentor Bot

Current users (#1):

#	Name	User Id	Created	Email	Actions
1	admin	275493460	Jan/29/2014	TM_alerts@securityinnovation.com	edit user view activity

[back to commands list](#)

We can also double check on the *TBot's DebugInfo* page that the `UserData` now points to a different folder (note that the folder name is based on the repo name)

Debug Info	
TM Config values	
TM.Xml.Database.Current.Path_XmlDatabase	: C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\App_Data\Libr
ary_Data.XmlDatabase	
TM.Xml.Database.Current.Path_XmlLibraries	: C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\App_Data\Libr
ary_Data.XmlDatabase\TM_Libraries_Test	
TM.UserData.Current.Path_UserData	: C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\App_Data\Libr
ry_Data.XmlDatabase\User_Data_Git_Site_TM_34_QA_Azure	
TM.UserData.Current.Path_UserData_Base	: C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\App_Data\Libr
ary_Data.XmlDatabase\User_Data	
TM.UserData.Current.Git_UserData	:
AppDomain.CurrentDomain.BaseDirectory	: C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\
02_Temp_Dir	: C:\DWASFiles\Sites\tm-34-qa\LocalAppData\02_5.1.1.02.Temp_1_29_2014

We can also confirm in the [DinisCruz-Dev/Site_TM_34_QA_Azure](#) repo that the UserData default files have been created (which also confirms that the connection between TeamMentor and Git's UserData repo is working ok)

User Data repo for <https://tm-34-qa.azurewebsites.net> website — Edit

6 commits 1 branch 0 releases 1 contributor

branch: master Site_TM_34_QA_Azure

changed: Users/admin_aea...(userData.xml)

unknown-user authored just now latest commit 9c4e6862b9

Users changed: Users/admin_aea... just now

README.md Create README.md 8 minutes ago

TMSecretData.config Added: TMSecretData.config , Users/admin_aea... 3 minutes ago

README.md

Site_TM_34_QA_Azure

User Data repo for <https://tm-34-qa.azurewebsites.net> website

Step 5: Create a new user and confirm that it shows up in GitHub

If you look at the Users folder in the GitHub repo, you should see one file in there (which represents the default admin user)

PRIVATE DinisCruz-Dev / Site_TM_34_QA_Azure

branch: master Site_TM_34_QA_Azure / Users

changed: Users/admin_aea...(userData.xml)

unknown-user authored just now latest commit 9c4e6862b9

admin_aea... changed: Users/admin_aea...userData.xml just now

Next, create a new user (using for example the form provided in the home page of the target TeamMentor site):

Login required

If you have a valid account, please [Login](#), if not, please sign up for an account to gain access:

Username:	<input type="text" value="dinus"/>
Password:	<input type="password" value="*****"/>
Repeat Password:	<input type="password" value="*****"/>
E-Mail:	<input type="text" value="dcruz@securityinnova"/>
First Name:	<input type="text"/>
Last Name:	<input type="text"/>
Company:	<input type="text"/>
Title/Job:	<input type="text"/>
Country:	<input type="text"/>
State:	<input type="text"/>

[Sign Up](#)

Once the account is created:

Selected Article 

Account created successfully

Please [Login](#)

Go back to TBot and reload the UserData (which will trigger a Git Pull and Push of the UserData repo):

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push) Done: true

Reload the GitHub User's folder and notice that there are two xml files in there:

PRIVATE [DinisCruz-Dev / Site_TM_34_QA_Azure](#)

branch: master [Site_TM_34_QA_Azure / Users](#) History

Added: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml
unknown-user authored just now latest commit 046247d6c8

..

changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml
admin_aeaf3806-88af-471e-a878-0463d9c69c82 authored 2 minutes ago

Added: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml
dinus_65cf66d2-bfba-4ea1-845b-6586e266562d authored just now

A quick look at the commits of this repo, will also show the Commits created by TeamMentor's backend:

PRIVATE [DinisCruz-Dev / Site_TM_34_QA_Azure](#)

branch: master [Site_TM_34_QA_Azure / Commits](#)

Jan 29, 2014

Added: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml
unknown-user authored a minute ago 046247d6c8

changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml
unknown-user authored 2 minutes ago 9c4e68629

changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml
unknown-user authored 3 minutes ago ea3794c045

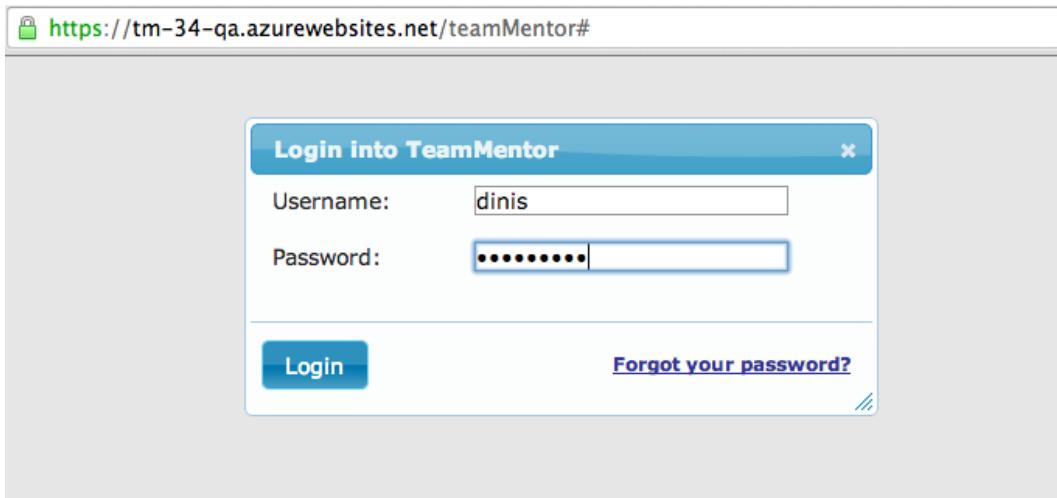
changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml
unknown-user authored 5 minutes ago 63cc42f710

changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml
unknown-user authored 5 minutes ago 2bc4ecae3

Added: TMSecretData.config , Users/admin_aeaf3806-88af-471e-a878-0463...
unknown-user authored 5 minutes ago 25086dd09a

Create README.md
DinisCruz-Dev authored 10 minutes ago 4b65666159

Now logout the admin user and login as the new **dinus** user:



... reload the UserData objects:

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push) Done: true

... and note in UserData GitHub's repo that there are a number of new commits:

Jan 29, 2014

- changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml (91bad035f9) - unknown-user authored just now
- changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml (53b91145d3) - unknown-user authored just now
- changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml (b7b1cb142c) - unknown-user authored just now
- changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml (038b349d89) - unknown-user authored a minute ago
- changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml (e2401fc296) - unknown-user authored a minute ago
- changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml (6325e2db7c) - unknown-user authored 7 minutes ago
- changed: TMSecretData.config (0c101fcdae) - unknown-user authored 9 minutes ago
- changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml (f6c8263040) - unknown-user authored 10 minutes ago

For example, here is the ‘Logout user activity log’ for the admin user:

changed: Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml

master

unknown-user (authored a minute ago)

1 parent e2401fc commit 038b349d891bd0feb30894b1b0641d7e60135e9d

Showing 1 changed file with 1 addition and 0 deletions.

1	Users/admin_aeaf3806-88af-471e-a878-0463d9c69c82.userData.xml	View
10	@@ -10,4 +10,5 @@	
11	<UserActivities Action="User Logout" Detail="admin" Who="admin" IPAddress="82.47.66.142" When="130354764	
12	<UserActivities Action="User Login" Detail="admin" Who="admin" IPAddress="82.47.66.142" When="130354764	
13	+ <UserActivities Action="View Article Html" Detail="A Centralized Log Server Is Deployed" Who="admin" IP	
14	</TMUser>	

0 notes on commit 038b349

[Show line notes below](#)

... and here is the UserActivity for the *dinis* user:

```

changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml
Browse code
master
unknown-user authored a minute ago
1 parent 53b9114 commit 91bad035f986bec22463dea644b385337d82e87f
Showing 1 changed file with 1 addition and 0 deletions.
Show Diff Stats
View
1  Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml
2  @@ -7,4 +7,5 @@
3  7    <UserActivities Action="Password Change" Detail="dinus" Who="dinus" IPAddress="82.47.66.142" When="1303
4  8    <UserActivities Action="User Login" Detail="dinus" Who="dinus" IPAddress="82.47.66.142" When="130354772
5    <UserActivities Action="View Article Html" Detail="A Centralized Log Server Is Deployed" Who="dinus" IP=
6    10   + <UserActivities Action="View Article Html" Detail="A Centralized Log Server Is Deployed" Who="dinus" IP=
7    11   </TMUser>

```

Step 6: Add a Library from a GitHub Repo

In order to make this a working server, we need to update the SecretData config file:

Available TBot Commands			
Welcome to the TBot control center, please chose the command to execute:			
Admin	Import Legacy Users	Reload Server Objects	UserData FirstScript
Admin\Config	Edit GitUserLocation	Edit SecretData	Edit TMConfig
Admin\LiveData	ActiveSessions	DebugInfo	IISSessions

With the location of the Library GitHit repo (see [Using TeamMentor 3.4 TBot admin pages to load and sync a Library hosted on GitHub⁵⁵](#) for a detailed explanation of its origin)

⁵⁵<http://blog.diniscruz.com/2014/01/using-teammendor-34-tbot-admin-pages-to.html>

The screenshot shows a web browser window with the URL https://tm-34-qa.azurewebsites.net/rest/tbot/run/Edit_SecretData. The page title is "TBot - your friendly TeamMentor Bot". Below it, a section titled "Editing an Secret Data" contains a JSON configuration object. The object includes fields for "Libraries_Git.Repositories" (containing two GitHub URLs), "Rijndael_IV" (a base64 string), "Rijndael_Key" (another base64 string), "SMTP_Password" (null), "SMTP_Server" ("smtp.sendgrid.net"), and "SMTP_UserName" ("TeamMentor"). A "Save" button is visible on the right, with a message "data loaded" displayed next to it.

```
{  
  "Libraries_Git.Repositories": ["https://DinisCruz-  
Dev:c78fa4d5dcf1b9f521a99396d667a00297734a2b@github.com/DinisCruz-  
Dev/Lib_.NET_3.5.git"],  
  "Rijndael_IV": "VPNm2Z01Zt4b6lY9M+ggcQ==",  
  "Rijndael_Key": "zr6ClPhJjlLiyggeeYcaEZgCtSIS2m7ydY2B8ga0EtU=",  
  "SMTP_Password": null,  
  "SMTP_Server": "smtp.sendgrid.net",  
  "SMTP_UserName": "TeamMentor"  
}
```

Next reload the Cache:

The screenshot shows the "TBot - your friendly TeamMentor Bot" interface again. It features a "Reload User Data (and Git Pull and Push)" button, a "Reload TMConfig" button, and a prominent "Reload Cache (will reload entire Xml database, including User data)" button, which is highlighted with a blue border. To the right of the cache button, the text "Reloading User Data" is visible.

On completion you should see a message with the number of Libraries and Articles in the current server:

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push)

Reload TMConfig

Reload Cache (will reload entire Xml database, including User data) Done: "In the library 'C:\\DWASFiles\\Sites\\tm-34-qaa\\VirtualDirectory0\\site\\wwwroot\\App_Data\\Library_Data\\XmlDatabase' there are 1 library(ies), 54 views and 1780 GuidanceItems"

And reloading TeamMentor will show the imported Library fully loaded and ready to be used:

The screenshot shows the TeamMentor web application interface. At the top, there's a navigation bar with links for Edit Mode, Change Password, Control Panel, Logout, TM 3.4, and a user logged in as admin. Below the navigation is the TeamMentor logo and a security innovation badge.

Applied Filters (with a clear button):

- Guidance Libraries** (selected):
 - .NET 3.5
 - Fundamentals of Security
 - OWASP Top 10
 - PCI DSS Code Review
 - PCI DSS Compliance
 - Security Engineering
 - Top 5 Rich Client Vulnerabilities
 - Top 5 Web Service Vulnerabilities

Search input field with a magnifying glass icon.

Technology filter checkboxes:

- Any
- .NET 3.5
- ASP.NET 3.5
- WCF 3.5
- Web Application

Phase filter checkboxes:

- Deployment
- Design
- Implementation
- Test

Type filter checkboxes:

- Attack
- Checklist Item
- Code Example
- Guideline
- How To
- Inspection Question
- Principle

Selected Article: **A Centralized Log Server Is Deployed**

Applies to (PCI DSS Requirements):

- PCI DSS Requirement 10.5
- PCI DSS Requirement 10.5.1
- PCI DSS Requirement 10.5.2
- PCI DSS Requirement 10.5.3
- PCI DSS Requirement 10.5.4
- PCI DSS Requirement 10.6
- PCI DSS Requirement 10.6.a
- PCI DSS Requirement 10.6.b
- PCI DSS Requirement 10.7
- PCI DSS Requirement 10.7.a
- PCI DSS Requirement 10.7.b

What to Check For: A centralized log server should be deployed.

Adding files to TeamMentor's web root via a **UserData** folder (synced with GitHub)

This post shows how to add custom files to the TeamMentor's webroot using a special feature of the TeamMentor's **UserData** folder.

In this demo I'm going to use the [UserData setup in this post⁵⁶](#) (currently synchronised with a GitHub repo)

Basically we are going to edit a file in GitHub, which will end up in the root of the associated TeamMentor website (which is quite a powerful PoC and bug fixing feature).

First step is to go to the synced GitHub repo ([created here⁵⁷](#)) and click the **Create a new file here** button in GitHub's UI:

User Data repo for <https://tm-34-qa.azurewebsites.net> website — Edit

18 commits 1 branch 0 releases

branch: master Site_TM_34_QA_Azure / [Create a new file here](#)

changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml

... the next page allows us to define a folder name (which needs to be **WebRoot_Files** if we want these files to be copied to webroot of the current TeamMentor application) :

PRIVATE DinisCruz-Dev / Site_TM_34_QA_Azure

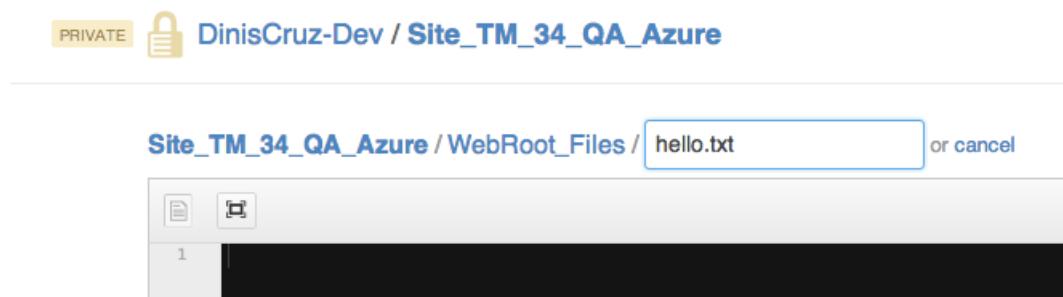
Site_TM_34_QA_Azure / or cancel

1

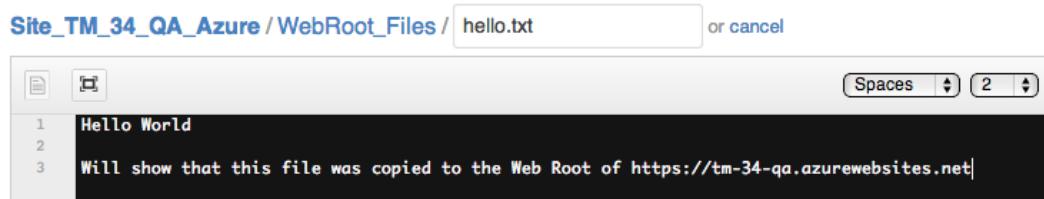
... and a file name (which can be anything):

⁵⁶<http://blog.diniscruz.com/2014/01/using-teammendor-34-tbot-admin-pages-to.html>

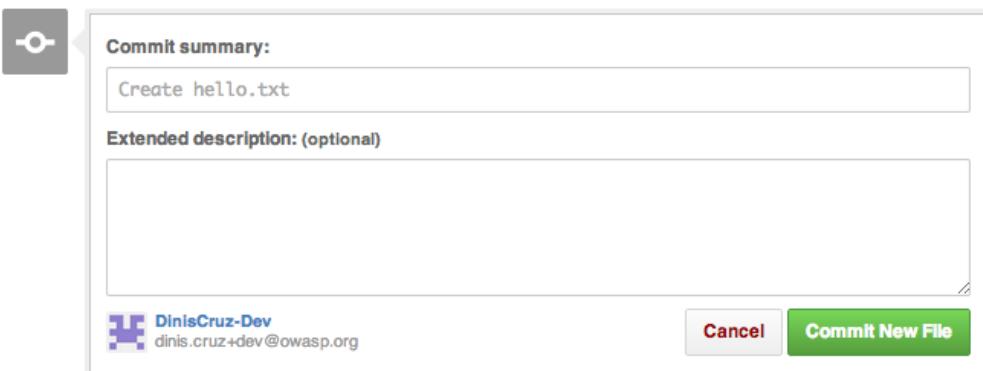
⁵⁷<http://blog.diniscruz.com/2014/01/using-teammendor-34-tbot-admin-pages-to.html>



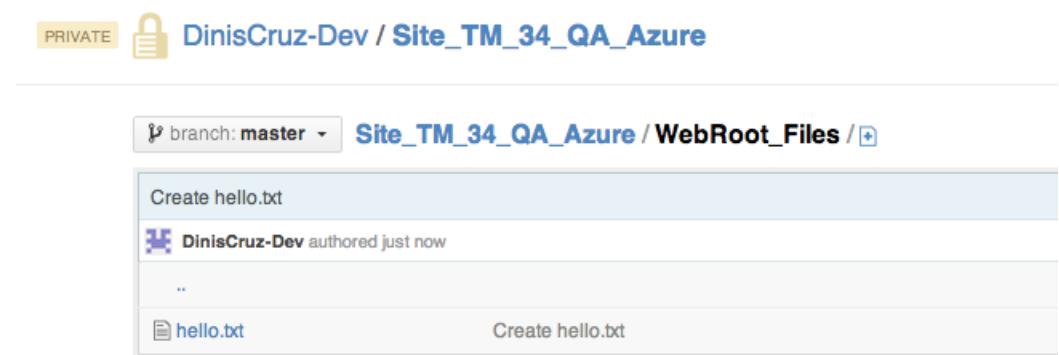
... the file contents are added using the GitHub's text editor:



... and saved using the *Commit New File* button:



Here is the file added to the GitHub's *UserData* repo:



Here is the commit (created by GitHub)

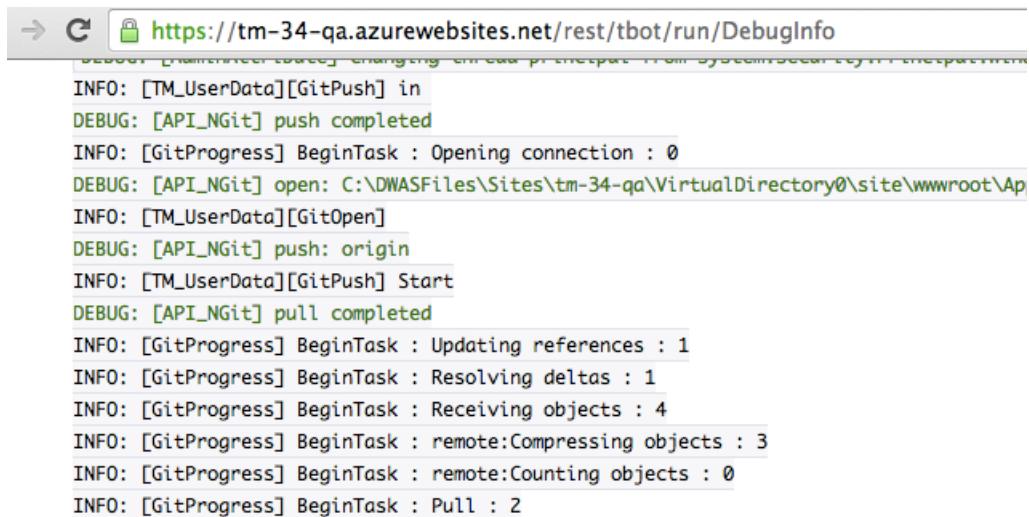
A screenshot of the GitHub commit history for the repository 'Site_TM_34_QA_Azure'. It shows three commits made on Jan 29, 2014:

- Create hello.txt** (DinisCruz-Dev authored just now) - Commit hash: c847b610a6
- changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml** (unknown-user authored 5 minutes ago) - Commit hash: 91bad035f9
- changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml** (unknown-user authored 5 minutes ago) - Commit hash: 53b91145d3

Back in TeamMentor, if we click on the Reload UserData

A screenshot of the TeamMentor interface. At the top, it says 'TBot - your friendly TeamMentor Bot'. Below that, there's a section titled 'Reloading Server objects' with a button labeled 'Reload UserData (and Git Pull and Push)' which is highlighted in blue. To the right of the button, it says 'Done: true'.

A server side (to TeamMentor) git pull will occur, and the file added in the GitHub's UI is now also present in the local TeamMentor's *UserData* folder:



The screenshot shows a browser window with the URL <https://tm-34-qa.azurewebsites.net/rest/tbot/run/DebugInfo>. The page displays a log of Git operations. The log entries are color-coded by category:

- INFO: [TM_UserData][GitPush] in
- DEBUG: [API_NGit] push completed
- INFO: [GitProgress] BeginTask : Opening connection : 0
- DEBUG: [API_NGit] open: C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\Ap
- INFO: [TM_UserData][GitOpen]
- DEBUG: [API_NGit] push: origin
- INFO: [TM_UserData][GitPush] Start
- DEBUG: [API_NGit] pull completed
- INFO: [GitProgress] BeginTask : Updating references : 1
- INFO: [GitProgress] BeginTask : Resolving deltas : 1
- INFO: [GitProgress] BeginTask : Receiving objects : 4
- INFO: [GitProgress] BeginTask : remote:Compressing objects : 3
- INFO: [GitProgress] BeginTask : remote:Counting objects : 0
- INFO: [GitProgress] BeginTask : Pull : 2

Note: in the next version of TM, the Git messages are much better (for example they will show the names of the files affected by a pull)

Here is a simple C# script that confirms that the file is already in the local *UserData* folder (this script was executed in the C# REPL that is part of TeamMentor's admin features)

Web C# REPL is a simple wrapper on O2 Platform's FluentSharp.CoreLib API
Note that all your script executions are logged (see link on top) so don't put any sensitive data here :)

```
1 var userDataPath = TM_UserData.Current.Path_UserData;
2 return userDataPath.files("*.txt",true);
3
4 //using TeamMentor.CoreLib;
5 //O2Ref:TeamMentor.CoreLib.dll
```

[Execute](#)

[C:\DWASFiles\Sites\tm-34-qa\VirtualDirectory0\site\wwwroot\App_Data\Library_Data\XmlDatabase\User_Data_Git_Site_TM_34_QA_Azure\WebRoot_Files\hello.txt]

But, at this stage, if we try to open the *hello.txt* file in a browser, we will see that it doesn't (yet) exist:



The reason is because the logic that checks for the existence of *WebRoot_Files* in the current *Userdata* folder, is only executed on server startup or cache reload.

The best solution is to go to the *TBot's Reload Server Object*'s page and click on the *Reload Cache* button:

TBot - your friendly TeamMentor Bot

Reloading Server objects

Reload UserData (and Git Pull and Push) Done: true

Reload TMConfig Done: true

Reload Cache (will reload entire Xml database, including User data) Reloading UserData

And once that is done, the *hello.txt* file will now exist in the TeamMentor's root folder:

← → C <https://tm-34-qa.azurewebsites.net/hello.txt>

Hello World

Will show that this file was copied to the Web Root of https://tm-34-qa.azurewebsites.net

Just to confirm that all is ok, let's try renaming that file in GitHub:

branch: master | Site_TM_34_QA_Azure / WebRoot_Files / hello.txt

DinisCruz-Dev 4 minutes ago Create hello.txt

1 contributor

file | 4 lines (2 sloc) | 0.103 kb

1 Hello World
2
3 Will show that this file was copied to the Web Root of https://tm-34-qa.azurewebsites.net

Open Edit Raw Blame History Delete

... to *helloAgain.txt*

Site_TM_34_QA_Azure / WebRoot_Files / helloAgain.txt

Code Preview

Hello World (Again)
Will show that this file was copied to the Web Root of https://tm-34-qa.azurewebsites.net

Spaces 2 No wrap

... saving it:



... checking that update commit is there:

Author	Commit Message	Hash	Action
DinisCruz-Dev	Update and rename hello.txt to helloAgain.txt	60f2225593	Browse code
DinisCruz-Dev	Create hello.txt	c847b610a6	Browse code
unknown-user	changed: Users/dinis_65cf66d2-bfba-4ea1-845b-6586e266562d.userData.xml	91bad035f9	Browse code

... reloading the TeamMentor's cache:

TBot - your friendly TeamMentor Bot

Reloading Server objects

- Reload UserData (and Git Pull and Push)** Done: true
- Reload TMConfig** Done: true
- Reload Cache (will reload entire Xml database, including User data)** Reloading UserData

... and finally confirming that the file has been updated in the live TM server.

A screenshot of a web browser window. The address bar shows a secure connection to <https://tm-34-qa.azurewebsites.net/helloAgain.txt>. The page content displays the text "Hello World (Again)" followed by a note: "Will show that this file was copied to the Web Root of https://tm-34-qa.azurewebsites.net".

Note: Doing this for ***.txt** file is not that interesting.

Where this technique will really show its power, is when we create ***.aspx** server-side pages, ***.html** client-side pages, ***.ashx** asp.net handlers or ***.cshtml** Razor pages (these last ones will need to be placed inside a special **TBot** folder).

I will show how this works in one of my next blog posts.

13. February 2014

- Reverting changes mades to TeamMentor articles

Reverting changes mades to TeamMentor articles

The problem was simple, there were a number of commits made to an TeamMentor GitHub repo that I wanted to completely reverse (without re-writing history).

For reference this happened when I was doing some ‘Link fixing’ tests on a server that was configured to auto commit to GitHub (which meant that the option to do a pure *git reset -hard* was not available since it would break the TM server)

In this case, the last good commit was [e794cc839689dfc7915099d39972abde643a969d¹](https://github.com/tm-sme/Lib_.NET_2.0/commit/e794cc839689dfc7915099d39972abde643a969d) and the last bad commit was [c53002083e85673f9a4dd7e6dbd2a37bc7ff9e2f²](https://github.com/tm-sme/Lib_.NET_2.0/commit/c53002083e85673f9a4dd7e6dbd2a37bc7ff9e2f) (currently HEAD of master)

My first idea was to just do a git revert to the [e794cc839689dfc7915099d39972abde643a969d³](https://github.com/tm-sme/Lib_.NET_2.0/commit/e794cc839689dfc7915099d39972abde643a969d) which worked ok locally.

But I struggled to merge it with the master HEAD, because git was being too clever , since it realised that these two commits were compatible, and just fast-forwarded into the most recent one (vs doing a ‘reverse merge’)

Based on the [this answer⁴](#) from SO’s Revert multiple git commits⁵ question, the solution was to

a) clone the target repo and create a test branch

```
$ git clone git@github.com:DinisCruz-Dev/Lib_.NET_2.0.git ****$ cd Lib_.NET_2.0.git $ git checkout -b mergeTest
```

b) do a git reset hard into where I wanted to go:

```
$ git reset --hard e794cc839689dfc7915099d39972abde643a969d
```

c) then do a git reset soft into the current master

```
$ git reset --soft a8b755098884173a8f6eced1faddefc0c34a987e
```

¹https://github.com/tm-sme/Lib_.NET_2.0/commit/e794cc839689dfc7915099d39972abde643a969d

²https://github.com/DinisCruz-Dev/Lib_.NET_2.0/commit/c53002083e85673f9a4dd7e6dbd2a37bc7ff9e2f

³https://github.com/tm-sme/Lib_.NET_2.0/commit/e794cc839689dfc7915099d39972abde643a969d

⁴<http://stackoverflow.com/a/1470452/262379>

⁵<http://stackoverflow.com/questions/1463340/revert-multiple-git-commits>

d) use the gitk tool to confirm that the local changes (about to be committed) exist after the current HEAD commit of the mergeTest branch

```
$ gitk
```

e) committed the changes

```
$ git commit -m 'reverting back to e794cc839689dfc7915099d39972abde643a969d
commit'
```

f) checkout master branch and merged with mergeTest branch

```
$ git checkout master
```

```
$ git merge mergeTest
```

```
Updating c530020..a2b3297
Fast-forward
 _Attack/2d684518-be94-4454-8d3b-e57f025b0083.xml | 206 ++++++-----+
-----_ _Attack/36208a74-52f2-4a48-9ecf-4d032d845f2b.xml | 162 ++++++-----+
-----_ _Attack/4c053210-1a24-44c2-a3f9-f0cf5008eb3f.xml | 138 ++++++-----+
-----_ _Attack/5af411c1-4606-4f7e-920c-186af71436c5.xml | 130 ++++++-----+
-----_ _Attack/6ce03806-c25d-4dbc-8df0-3343085d31d0.xml | 74 ++++++-----+
-----_ _Attack/7dfbe9c9-481e-46d2-b1a5-9a776578d6c2.xml | 137 ++++++-----+
-----_ _Attack/adf5df06-2b67-4e2a-ace2-6d7060e0bd95.xml | 209 ++++++-----+
-----_ _Attack/d4b48303-d535-4549-90fc-474b99eff901.xml
| 180 ++++++-----+
--_ _Attack/dcf4e714-d7e2-4c7d-8609-6ab5bd309476.xml | 134 ++++++-----+
-----_ _ 9 files changed, 734 insertions(+), 636 deletions(-)
```

f) pushed into GitHub

```
$ git push origin master:master
```

g) in the TM website trigger a cache reload (which will also do a git pull from GitHub), opening the DebugInfo page will also show a Git Pull message

https://tm-34-qa.azurewebsites.net/rest/tbot/run/Reload_Server_Objects <https://tm-34-qa.azurewebsites.net/rest/tbot/run/DebugInfo>

h) finally to confirm that all is really the way it should be, I opened up the Pull Request page for the affected repo, and there are now 5 commits , but with 0 files changed:

https://github.com/tm-sme/Lib_.NET_2.0/compare/master...DinisCruz-Dev:master

The screenshot shows a GitHub pull request comparison between two branches: tm-sme:master and DinisCruz-Dev:master. At the top, there are buttons for 'Create pull request' and 'Edit'. Below that, a summary indicates 5 commits, 0 files changed, 0 comments, and 2 contributors. The commit list for Feb 10, 2014, includes the following entries:

Author	Changes	Commit ID
unknown-user	changed: Attack/6ce03806-c25d-4dbc-8df0-3343085d31d0.xml	f60cc7a
unknown-user	changed: Attack/2d684518-be94-4454-8d3b-e57f025b0083.xml , Attack/362...	d395023
unknown-user	changed: Attack/2d684518-be94-4454-8d3b-e57f025b0083.xml , Attack/362...	a8b7550
unknown-user	changed: Attack/d4b48303-d535-4549-90fc-474b99eff901.xml	c530020
DinisCruz-Dev	reverting back to e794cc8 commit	a2b3297

At the bottom, it says 'Showing 0 changed files with 0 additions and 0 deletions.' and has a 'Show Diff Stats' button.

Note how in the screenshot above, the last commit is the one done created during this blog post (which reverts the other 4).

There is probably a better way to do this, but the solution described above was the one that made more sense to me (and the one that worked :))