# Community Contribution Steps

Contents

[Community Contribution Steps 1](#_Toc474241959)

[Prequisites 1](#_Toc474241960)

[Useful Information 4](#_Toc474241961)

[CC Workflows 5](#_Toc474241962)

[Steps to handle a new PR 6](#_Toc474241963)

[Alternative Steps 11](#_Toc474241964)

[Steps to close an existing PR 13](#_Toc474241965)

[When it was accepted. 13](#_Toc474241966)

[When it was Rejected. 14](#_Toc474241967)

[When it was partially approved. 14](#_Toc474241968)

[Merging Public Repos 17](#_Toc474241969)

[Action Items 18](#_Toc474241970)

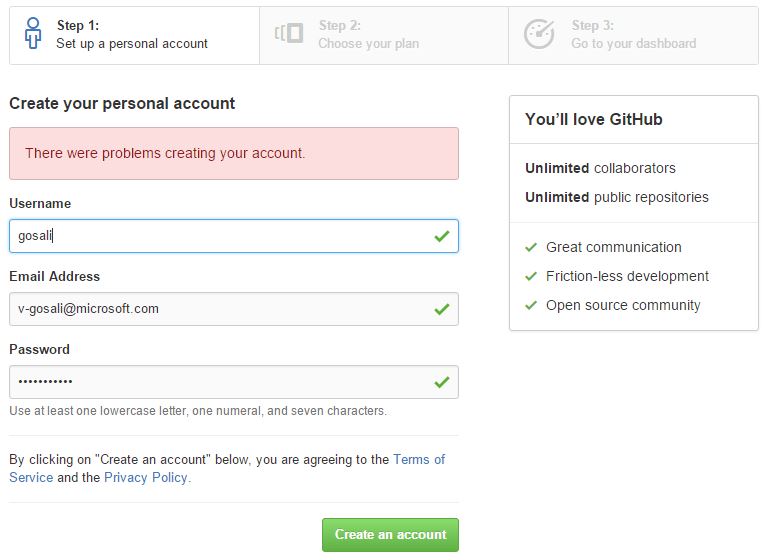
## Prequisites

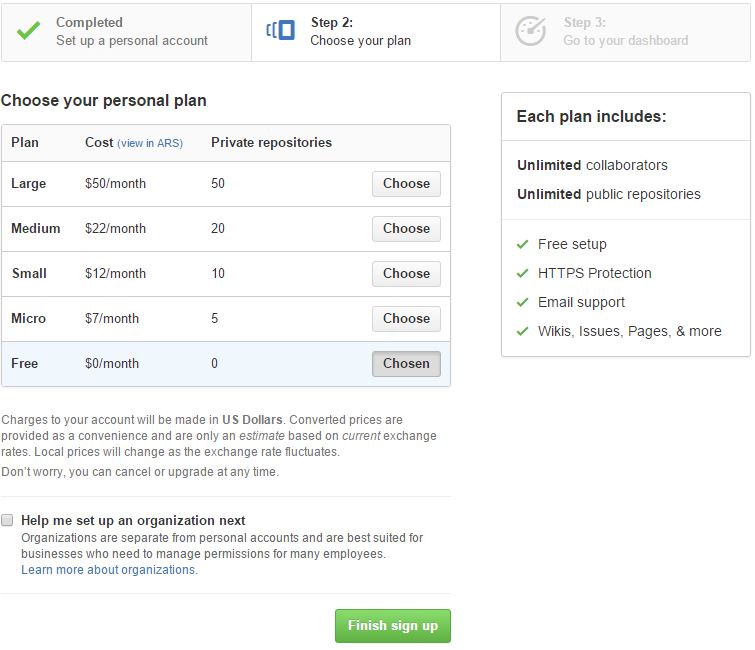
1. Go to <https://github.com/>
2. Click on **“Sign Up for GitHub”**
3. Complete the next information:  
   **Username:** your V-account name

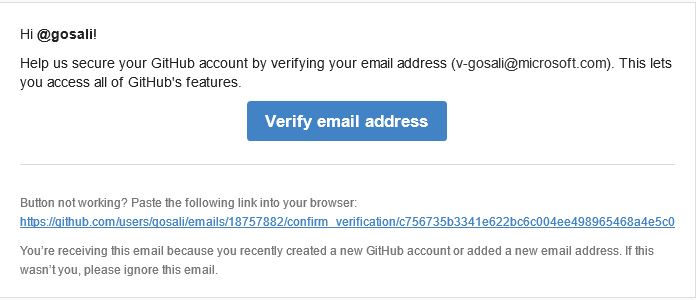
**E-mail:** your Microsoft e-mail

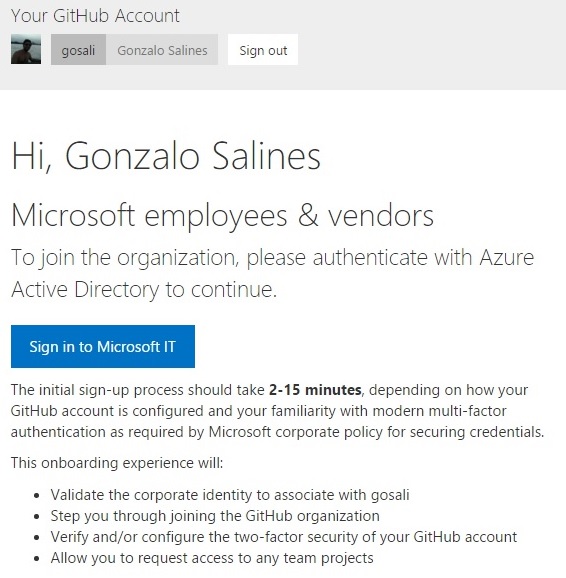
**Password:** \*\*\*\*\*\*\*

Example SS:



1. Click on **“Create an account”**
2. Select the Free Plan and click on **“Finish Sign Up”**
3. You will receive an e-mail to verify your email address. Do it!



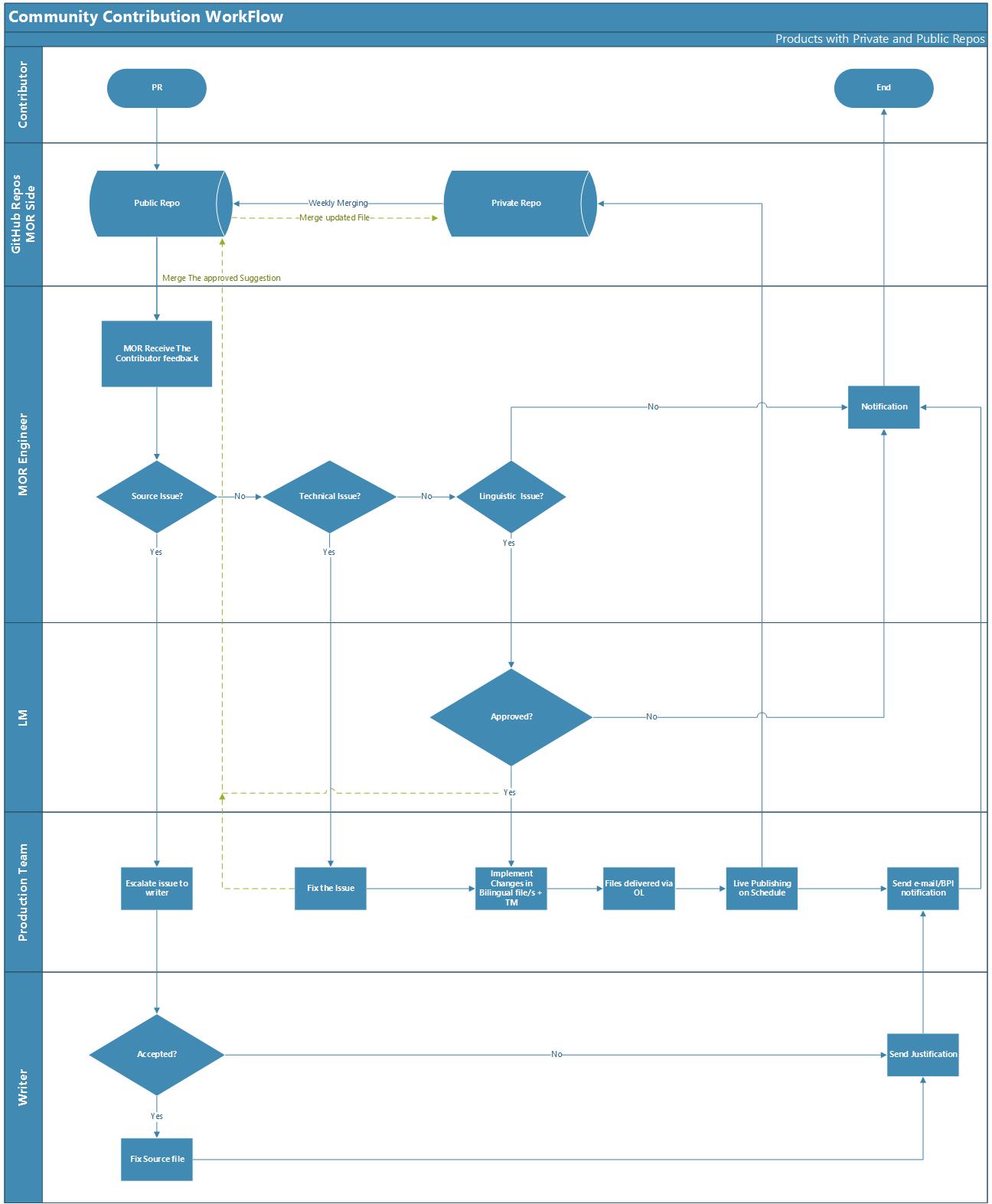
1. Now you have your GitHub account created. Update your Profile if you need.
2. Once it’s done access to the next URL:  
   <https://azureopensource.azurewebsites.net/>
3. Select **“Use Application”**
4. In the next page click on **“Sign in to Microsoft IT”**
5. Request to Join to the Desire Repos/teams
6. Clone all Public Repos locally.  
   [<Repos to clone >](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Repos%20Links&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={452F5964-50E0-4C67-A94F-4443C5585411}&end)

## Useful Information

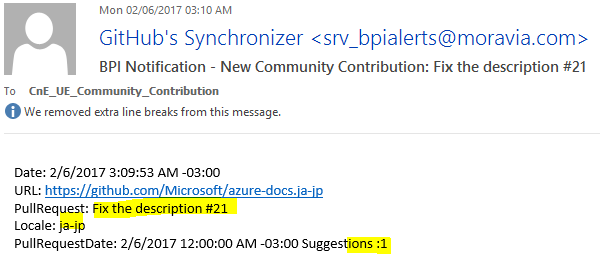
* [LMs Portal](https://portal.moravia.com/prj/msce/LinguisticServices/wiki/SitePages/Language%20Moderators'%20Contacts%20and%20Product%20Ownership.aspx)
* [MS Community Contribution OneNote](https://microsoft.sharepoint.com/teams/CEIntl/LocKit/_layouts/15/WopiFrame2.aspx?sourcedoc=%7Ba992d173-987a-4d2d-98b1-9adf856e154e%7D&action=edit&wd=target%28GitHub-DPS%2ECurrent%2Eone%7C1CB4385D-00D7-4787-94E0-6DF1547065DD%2FCommunication%20Scenarios%7CAD36521A-023B-47A0-852D-25A2A50E8863%2F%29): Please scan it to get the big Picture
* [Moravia CC OneNote](file:///\\parana\Phases5\Microsoft_CnE\UA\Phases\Community_Contribution\01_Documentation\CC_OneNote\Community%20Contribution\CC.one)
* [Use Cases:](file:///\\parana\Phases5\Microsoft_CnE\UA\Phases\Community_Contribution\10_UCases\) Extremely Useful. Read them for better understanding. Find them also in [<BPI>](https://bpi.moravia.com/Default.aspx#ViewID=UseCase_ListView&ObjectClassName=MORXAF02.Module.BPE.UseCase)
* [CC Tracking sheet.](file:///\\parana\Phases5\Microsoft_CnE\UA\Phases\Community_Contribution\03_Tracking_Sheet\CnE_Community_Contribution_Tracking_Sheet.xlsx)

## CC Workflows

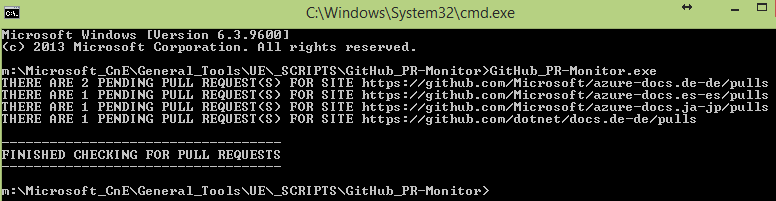
[<Path>](file:///\\parana\Phases5\Microsoft_CnE\UA\Phases\Community_Contribution\01_Documentation\CC-WorkFlow_13122016\)



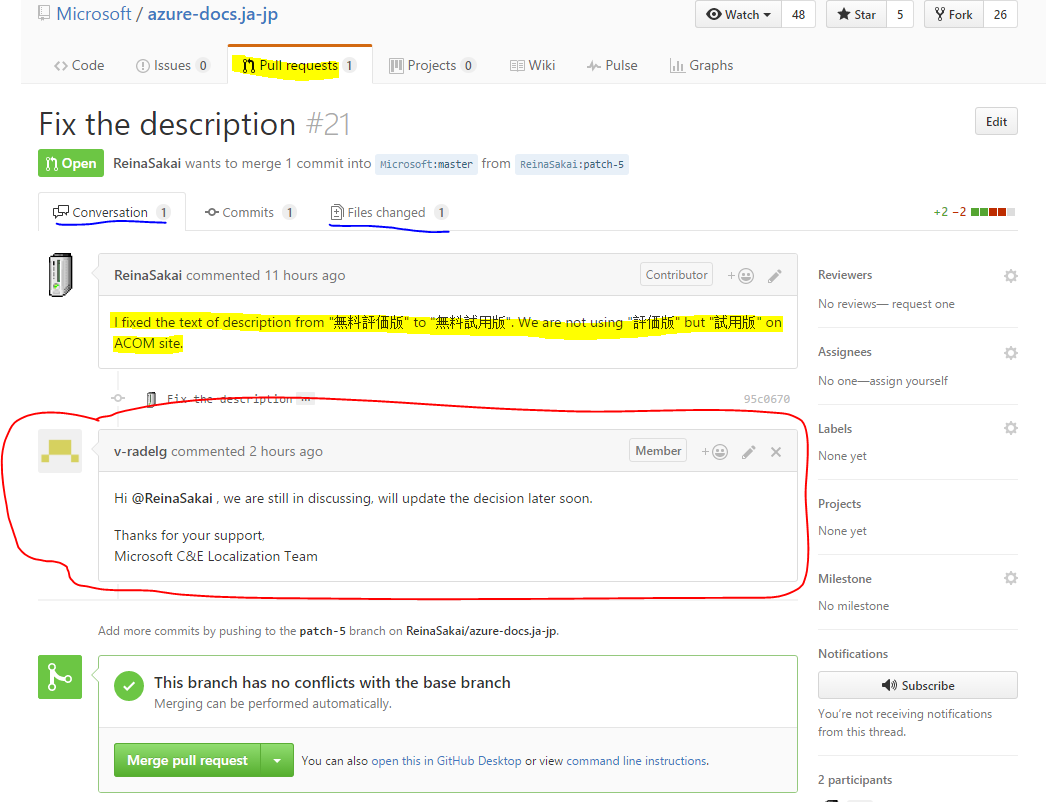
## Steps to handle a new PR

1. Check if there is a new PR. You can do it with two methods.
   1. If you are in the CC Outlook alias [CnE\_UE\_Community\_Contribution@moravia.com](mailto:CnE_UE_Community_Contribution@moravia.com) you will receive and e-mail notification.  
      

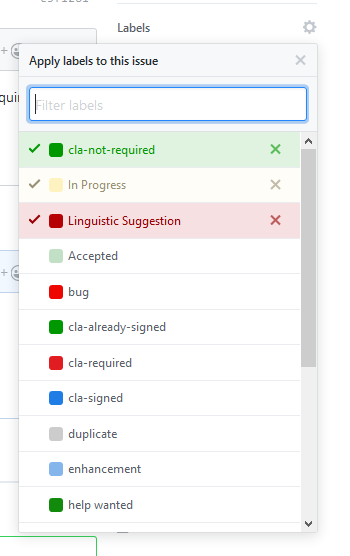
If you received this notification, is because the Contribution was partially imported to BPI.

* 1. Check new PR with the “[GitHub\_PR-Monitor.exe](file:///m:\Microsoft_CnE\General_Tools\UE\_SCRIPTS\GitHub_PR-Monitor\GitHub_PR-Monitor.exe)”. Run it with CMD.  
     

It will show you how many opened PRs are at the moment and their link.

1. Open the PR GitHub link.
2. Click on Pull Request. There you will see all the Opened PRs. Select one of the PRs and You will see What the contributor report, and how many Files have been changed. (it should be just one).  
     
   Add the comment from the [Communication scenarios](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Communication%20Scenarios%20&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={A51D9F7C-C3A9-4693-BE90-E9C3366CFE90}&end) list. The one to keep the contributor on the loop. (Important: ALWAYS answer the contributor in the language they use to comment)

Add corresponding labels to the PR



The labels depend on the status and type of PR. Due to the nature of some PRs, two or more types can coexist.

The usual labels selected by us are:

Inprogress

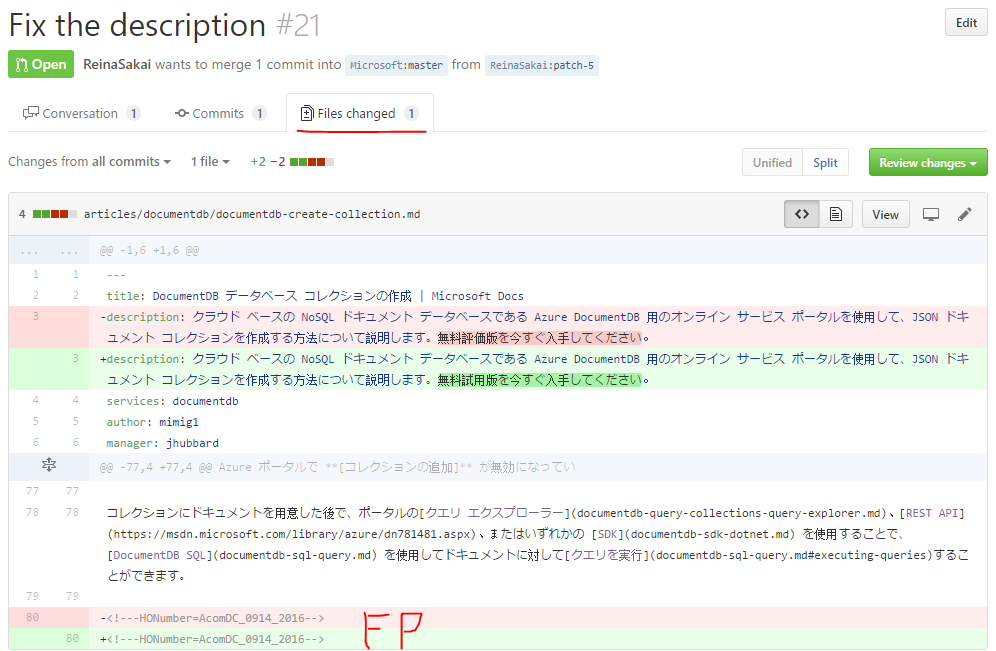
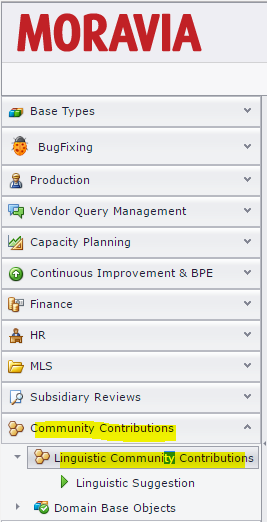
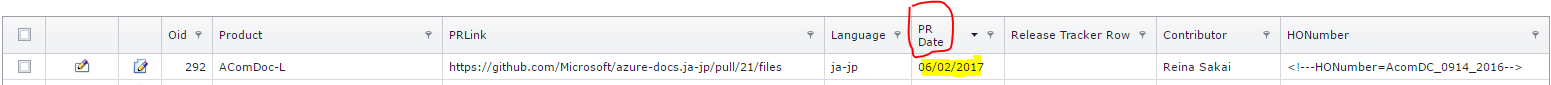
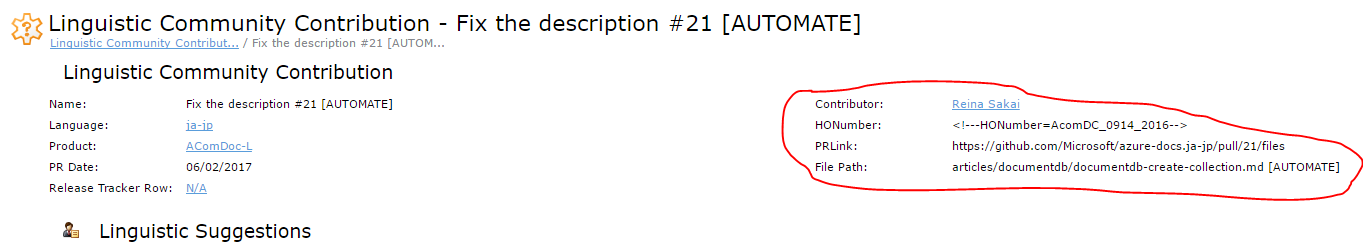
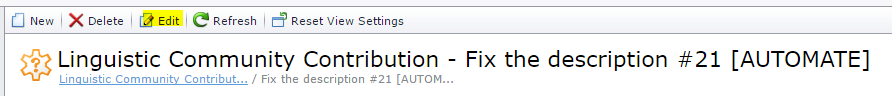
Accepted

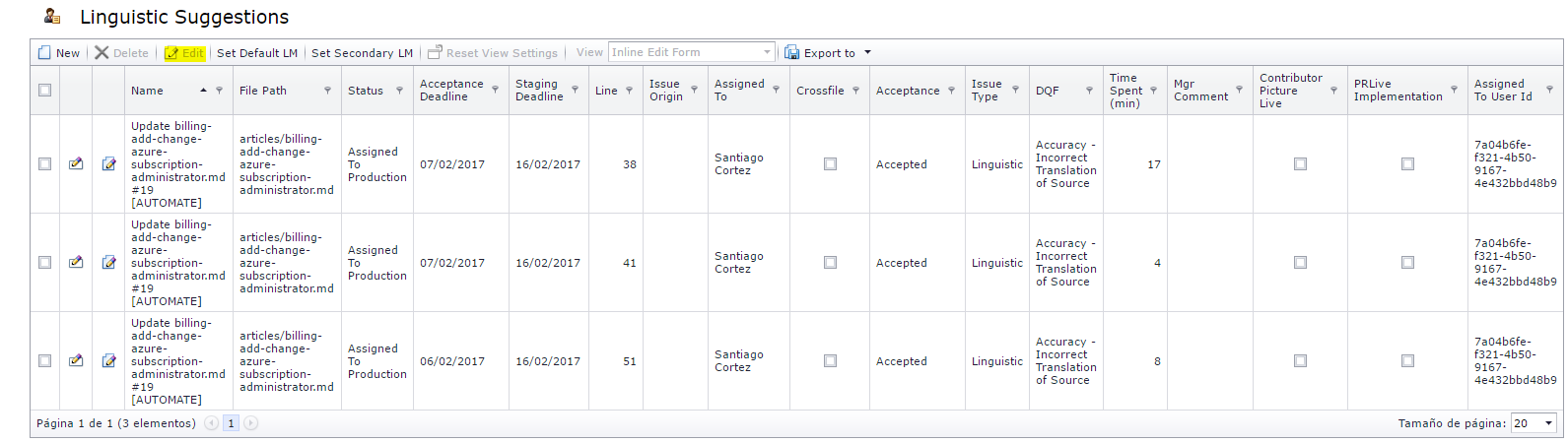
Rejected

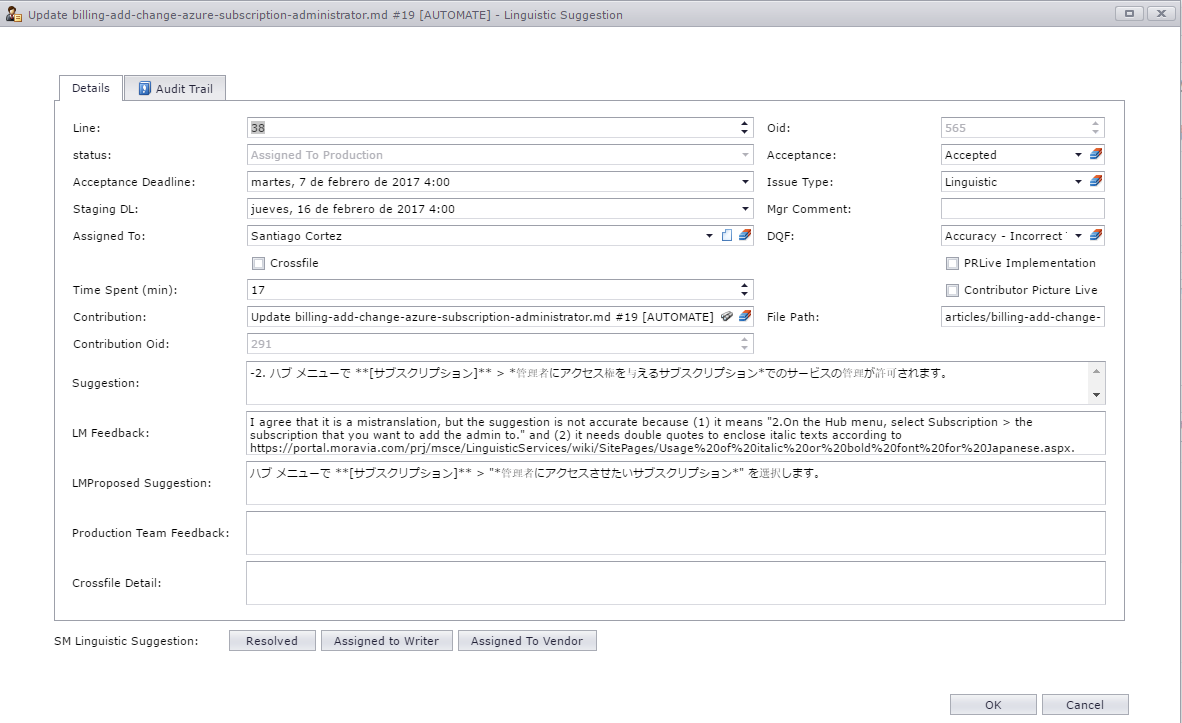
Liguistic suggestion

Technical issue

Source issue

1. Click on files changes to see what the suggestion is.  
   
2. We need to identify if it’s a Linguistic Suggestion, technical or source issue.  
   **Note:** Sometimes GitHub does not Highlight the changes correctly, and instead, it highlight the whole article. In that case you need to make the comparison using a different tool.
3. Once you have detected all the changes made by the contributor, access [BPI](https://bpi.moravia.com/)   
   
4. Go to “Linguistic Community Contributions” to see the auto-imported PR.  
   The “Linguistic Suggestion” Shows all the suggestions in a pool.
5. Order it by PR date.  
   
6. Access to the PR.  
   
7. Check that all fields were correctly imported. IF not, update it manually.
   1. Click on Edit  
      
   2. Fill/modify the corresponding fields
   3. Click “Save and close”
8. Now that the PR has been checked and it’s OK! Check the suggestions. They are in the grid view below.



1. Select a Suggestion and click on edit to check, and complete the remaining empty fields, or if it has something to modify.  
   

**Acceptance:** Whether the suggestion is accepted or not.

**DQF:** type of translation error the suggestion is about.

**Time spent:** How much time LM have spent on this task

**LM feedback:** Type any relevant feedback about the suggestion showed in the field above  
**LMProposed Suggestion:** If the suggestion is not correct and the original translation in the file is no correct either. LM will complete it with his/her own proposed suggestion. Also he/she will transcribe here the suggestion from the contributor if it’s is correct.

**Acceptance Deadline:** Is the deadline LM have to complete and share his/her feedback

1. Once everything is completed. Assign it to the correct person. (unless the suggestion is Invalid and needs to be rejected) If it’s Linguistic suggestion, select the corresponding LM. If it’s something Technical/source Issue, select someone from The production team that Handles that product.
2. Update the Status machine. (icons at the bottom)
   1. If it’s Linguistic suggestion: “AssignedToLM”
   2. If it’s something Technical/source Issue: “AssignedToProduction”.
   3. If it’s an Invalid Suggestion: “Rejected”

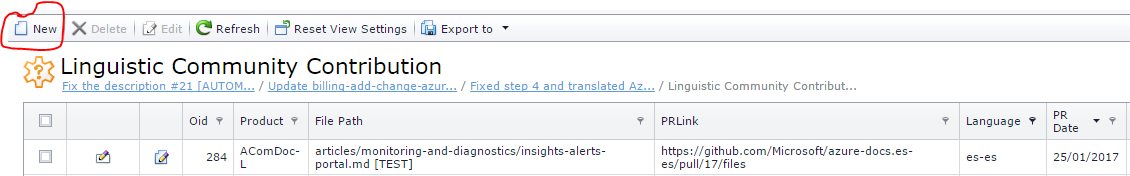
**Note:** An automatic e-mail alert will be sent to the assigned person, when saving and closing

1. You can also send an e-mail notification to the LM to make sure they will handle the PR.  
   [E-mail template path.](file:///\\parana\Phases5\Microsoft_CnE\UA\Phases\Community_Contribution\01_Documentation\Product%20%20Community%20Contribution%20%20PR# x  xx-xx 2.oft)
2. Update CC [Tracking sheet](file:///\\parana\Phases5\Microsoft_CnE\UA\Phases\Community_Contribution\03_Tracking_Sheet\CnE_Community_Contribution_Tracking_Sheet.xlsx).

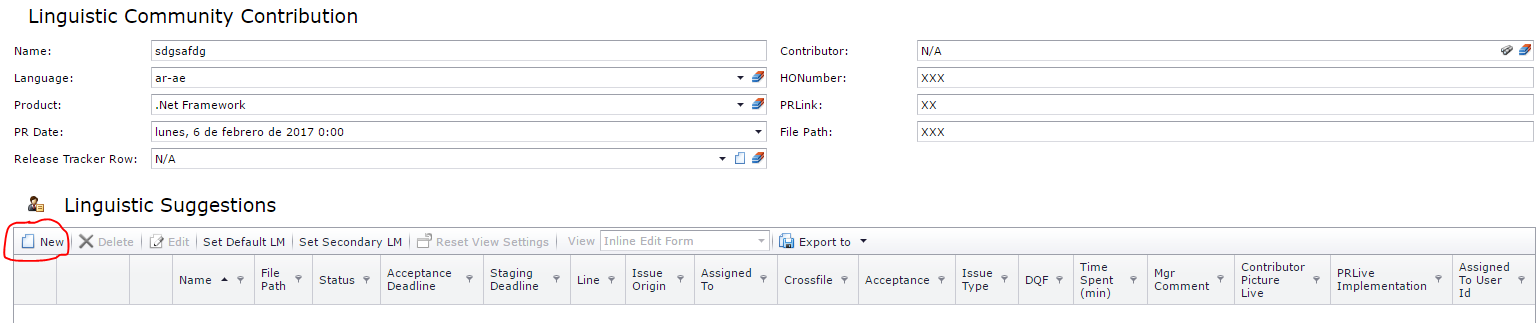
### Alternative Steps

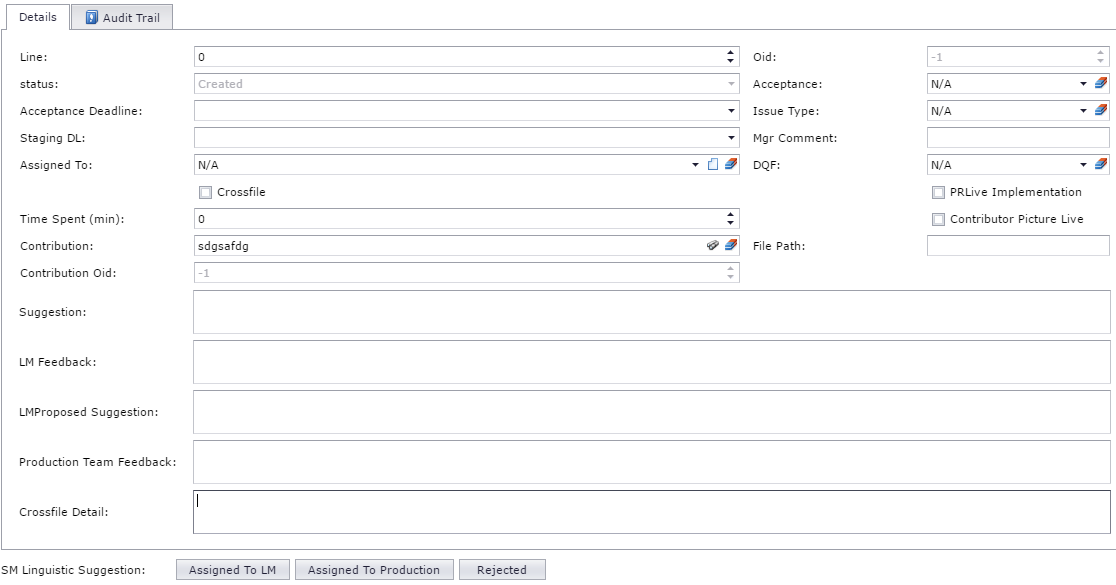
When the PR was not automatically imported.

1. Go to BPI. To the Community Contributions tab.
2. Click on Create new



1. Complete all fields there



1. Then click on “New” and create one ticket per suggestion.  
     
   **Line:** suggestion line number.

**status:** Auto populated

**Acceptance Deadline:** From the PR creation date add 3 business days

**Staging DL:** From the PR add 2 weeks. Then this will be changed by the Prod Team that handles the Publishing process.

**Assigned To:** The person who will receive the Notification. If it’s a Linguistic suggestion you will select the corresponding LM.

**Time Spent (min):** The LM will fill later this field with the time he/she has spent. The Prod team will use it for invoicing purposes

**Contribution:** Contribution Name (took it from GitHub)

**Suggestion:** Copy there the suggestion made by the contributor. And also the current one.

**LM Feedback:** will be filled by the LM

**LMProposed Suggestion:** will be filled by the LM

**Production Team Feedback:** will be filled by the Prod Team, or you can use it also to give some details about the suggestion, if needed

**Crossfile Detail:** Prod Team will detail here the files that have the same affected string.

**Acceptance:** LM/ProdTeam will selected either accepted or rejected.

**Issue Type:** Select Linguistic, technical or Source Issue

**Mgr Comment:** Manager comment if needed

**DQF:** LM will select type of linguistic issue.

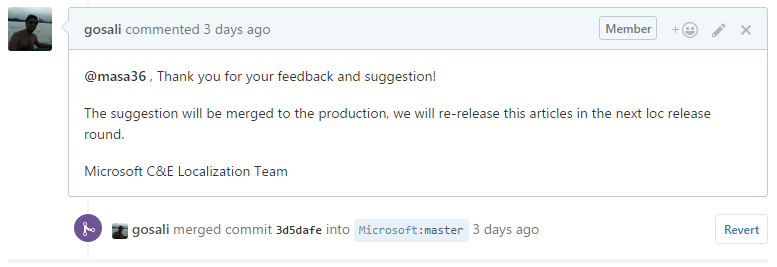
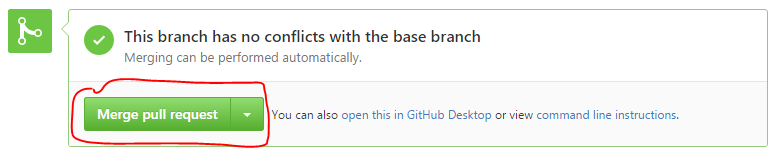
**File Path:** Dynamic path of the file in the repo including the file name

**SM Linguistic Suggestion:** Status machine. Select depending the suggestion the corresponding status

1. Click “Save and Close” or “Save and New” if there is more than one suggestion

## Steps to close an existing PR

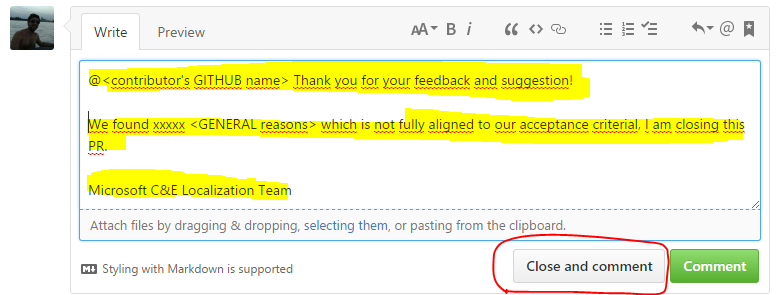
### When it was accepted.

1. Once the LM or the Production Team completed the BPI ticket. Open the PR link and reply with the corresponding [Communication scenarios](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Communication%20Scenarios%20&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={A51D9F7C-C3A9-4693-BE90-E9C3366CFE90}&end). (If the Contributor write in native language use the corresponding translated scenario in the OneNote)  
   
2. Merge the file to the Public repo. Click on “Merge Pull Request”  
   
3. Click on confirm.
4. After this we need to Merge this PR (actually the modified file) that is on the Public Repo to the Private Repo. This way the Contributor Portrait will appear in the website that correspond to the file merged.  
     
   **Note:** For A.com Doc the PR merging is Handled by Lucas Fernandez and Team. So No actions needed on our side.
5. Update BPI. Access to the Corresponding ticket. Check if LM has completed/update all necessary fields; if not, complete them. Update Status machine:
   1. If it’s Linguistic suggestion: “AssignedToProduction” And also assign to the corresponding person from the Prod Team.
   2. If it’s something Technical/source Issue: “Resolved”.
6. Update CC [Tracking sheet](file:///\\parana\Phases5\Microsoft_CnE\UA\Phases\Community_Contribution\03_Tracking_Sheet\CnE_Community_Contribution_Tracking_Sheet.xlsx).

**Important:** Never merge Source issues. They are always closed with the corresponding response.

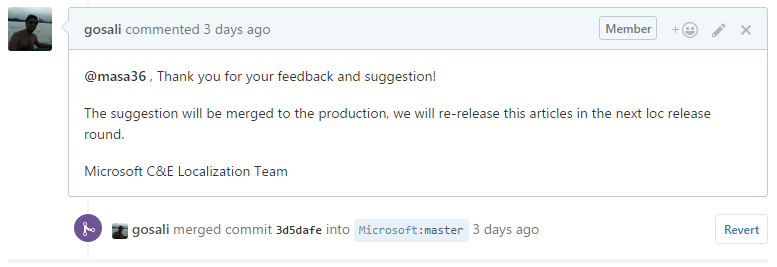
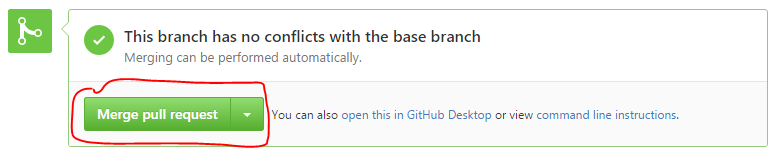
### When it was Rejected.

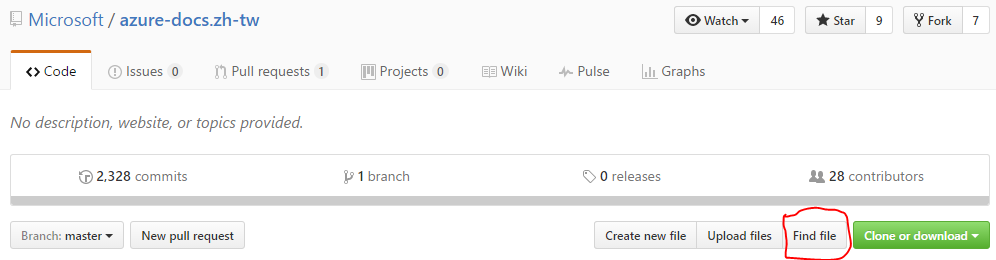
**Check OneNote to know the criteria to reject PRs.**

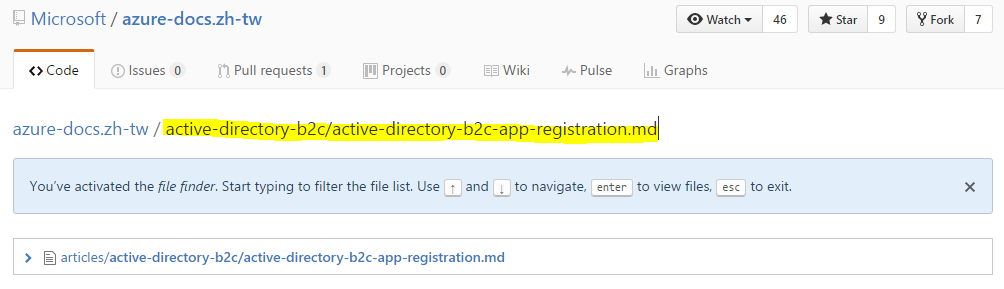
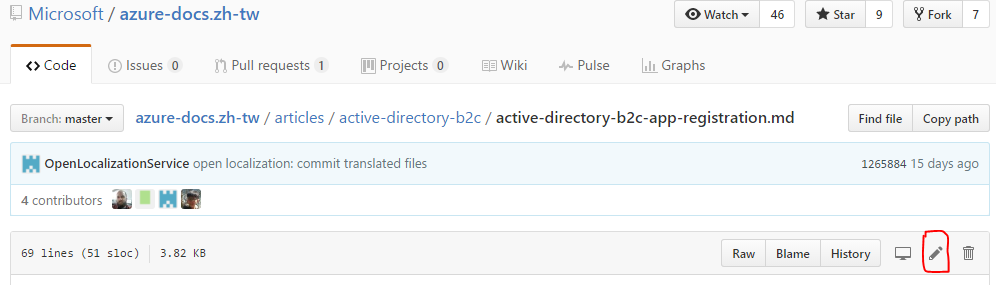
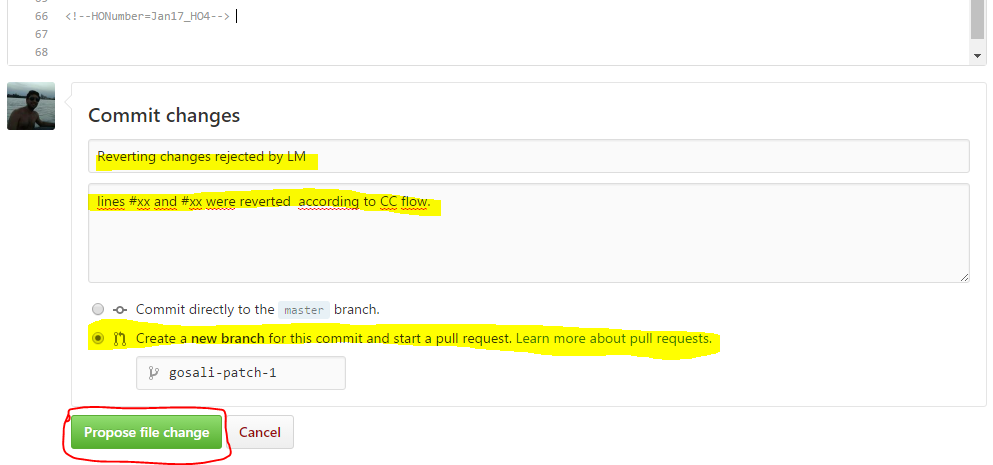
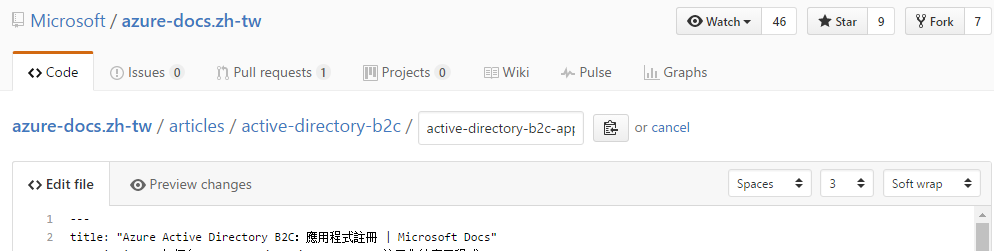
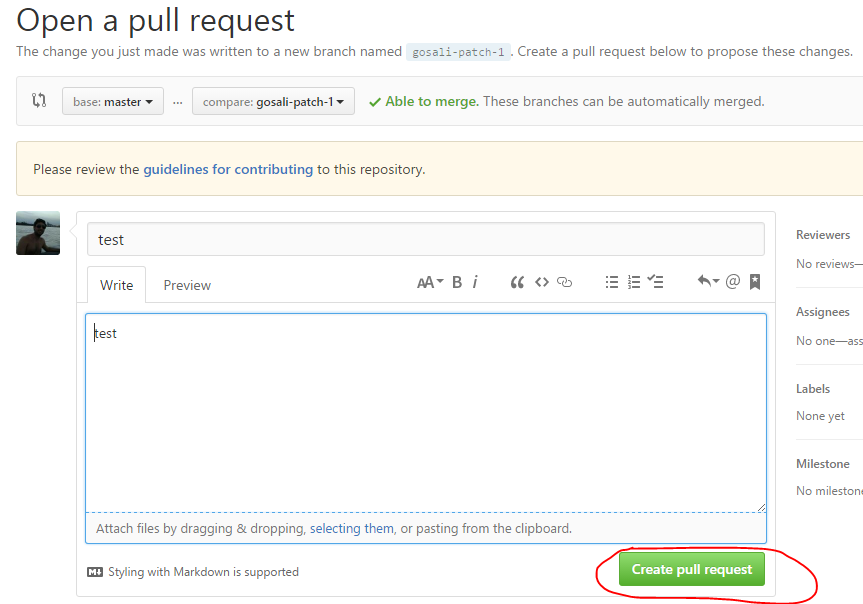
1. Once the LM or the Production Team completed the BPI ticket. Open the PR link and reply with the corresponding [Communication scenarios](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Communication%20Scenarios%20&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={A51D9F7C-C3A9-4693-BE90-E9C3366CFE90}&end). (If the Contributor write in native language use the corresponding translated scenario in the OneNote)
2. Click on Close the PR.  
   

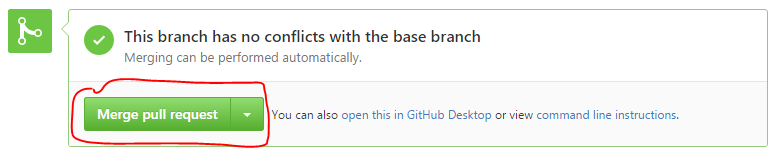
### When it was partially approved.

**When the PR has several suggestion and some of them were accepted and some are not.**

1. Once the LM or the Production Team completed the BPI ticket. Open the PR link and reply with the corresponding [Communication scenarios](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Communication%20Scenarios%20&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={A51D9F7C-C3A9-4693-BE90-E9C3366CFE90}&end). (If the Contributor write in native language use the corresponding translated scenario in the OneNote)  
   
2. Merge the file to the Public repo. Click on “Merge Pull Request”  
   
3. Click on “confirm”.
4. Go to the Repo main page in GitHub
5. Click on “Find File”



1. Search the file you have just merged. (that has the accepted and the rejected suggestions)  
     
   **Note:** Remember to include the sub-folder when searching because there might be more than one file with the same name.
2. Click on the file to open it.
3. Click the pencil icon to edit the file.  
   
4. Edit the file by changing back to the current strings, the lines that were modified by the contributor but rejected by the LM.
5. At the bottom of the file you will need to fill the title for the new PR that will revert the changes and also some description.
6. Click on “Propose file change”
7. Then click on “Create pull request”  
   
8. Merge the file to the Public repo. Click on “Merge Pull Request”



1. Click “Confirm”. And now The file will only have the accepted changes.

# Merging Public Repos

1. Open Desktop GitHub. Clone all public repos listed in the [OneNote](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Repos%20Links&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={452F5964-50E0-4C67-A94F-4443C5585411}&end).
2. Follow the [schedule](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Repo%20Merging%20|%20Weekly%20Schedule&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={8C86895F-BC47-4847-A276-68FEA257C274}&end) for Merging.
   1. Do not merge the reported Repos with conflicts.
3. Use the commands in the [OneNote](onenote://PARANA/Phases5/Microsoft_CnE/UA/Phases/Community_Contribution/01_Documentation/CC_OneNote/Community%20Contribution/CC.one#Github%20commands&section-id={3AD48EEA-D218-4EF0-8F92-0BB019FE7A00}&page-id={B2FAE98F-9BE9-4E11-818B-8349CE4E9096}&end) to perform the merging.

# Action Items

1. Monitor/Detect new PRs.
2. Update/create BPI tickets for new PRs.
3. Respond to the Contributors in GitHub.
4. Merge/close PRs in GitHub
5. Update TS.
6. Follow UP LMs and Prod teams and make sure they update BPI.
7. Merge Public repos.