[Skip Navigation Links](https://portal.moravia.com/prj/msce/LinguisticServices/partnerengagement/SitePages/Community%20Contribution%20(CC)%20Reviews.aspx#ctl00_PlaceHolderPageTitleInTitleArea_ctl00_SkipLink)[Community Contribution (CC) Reviews](https://portal.moravia.com/prj/msce/LinguisticServices/partnerengagement/SitePages/Community%20Contribution%20(CC)%20Reviews.aspx)

|  |
| --- |
| This section describes the Community Contribution (CC) review process and provides the LMs all the information needed to carry out this task.  Overview  End users provides linguistic suggestions and change requests in GitHub in the form of Pull Request (PR), which are sent to Moravia for triage and confirmation.  As the Moravia target language owner for Cloud and Enterprise Products, the LM's input and insight are needed to evaluate the Community feedback.  You will be asked to analyze the feedback and the errors reported and follow the below process for reporting to Moravia.  Overview of the process  This section provides an overview of the overall process, the steps for the LMs are detailed in the following sections.   1. End users enter linguistic suggestions / change requests in GitHub in the form of Pull Request (PR). 2. Moravia engineering team will create the BPI Pull Requests based on the request in GitHub and assign them to the LMs. 3. LMs receive a notification from BPI from engineering team. 4. Within the deadline provided in the BPI request, LMs confirm if the request from the end user should be applied or rejected, always referring to:    * MS TermStudio    * MS Style Guide    * [CnE Knowledge Portal](https://portal.moravia.com/prj/msce/LinguisticServices/wiki/_layouts/15/start.aspx#/SitePages/CnE%20Knowledge%20Portal.aspx)   And providing:   * + Issue type and taxonomy   + A clear and detailed linguistic justification for the rejection / acceptance   + Spent time for each task.  1. Production Teams will assign to vendor and implement approved changes in the file/s in OddJob. 2. Engineering team enters LMs feedback in GitHub. 3. Production Teams will invoice time spent for each task.   Handling the Pull Request in BPI  1. The engineering team create the Pull Request entry and assign it to the LM through BPI [https://bpi.moravia.com](https://bpi.moravia.com/).  The PR request in BPI will contain all the information from the GITHUB PR (PR name, language, Product and PR date) needed for the LM to triage the request.  Use your MNET account to log in in BPI.  CC_1.png     * **Name:** PR Name * **Language:** Language code * **Product:** Product Name * **PR date:** Date when the PR was created in GitHub * **Contributir:** User name who create the PR * **HOnumber:** HO unique ID of the file which has been modified. * **PRlink:** The link to the PR and modified file Extremy important to see the whole context and the Contributor explanation. * **File Path:** File name with dynamic Path from GitHub. * **Resolution:** PR resolution. It can be either “accepted” or “rejected” * **Status:** PR status   Each GitHub PR can contain one or more suggestions in one or more than one file. The LM receive a notification for each linguistic suggestion,  which must be treated separately in BPI. The following steps need to be performed on each Linguistic Suggestion received.  2. Once received the request, click on the click provide in the e-mail and will open the suggestion windows as below.    3. Then fill carefully all the following fields.   * **LM feedback:** LM feedback on the suggestion * **LMProposed Suggestion:** If neither the current and the proposed translation are correct. Please add which would be the correct one. * **Acceptance:** LM will have to select "Accepted" or "Rejected" accordingly * **Issue type:** LM will have to select "Linguistic", "Source Issue" or "Technical" accordingly. * **DQF:** LM will have to select the suggestion taxonomy accordingly * **Time Spent (min):** LM will have to report the time spent on resolving each suggestion   In addition, the PR will contain the following fields that don't need to be updated by the LM. But are important to check.   * **Contribution:** PR Name * **Contribution OId:** The ID for the contribution, in case you need to search it to know exactly the amount of suggestions created on that PR. * **Line:** corrected file line * **Cross-file:** Production Teams will check this if necessary. * **Acceptance deadline:** LM deadline to accept/reject PR * **Staging deadline:** Production team deadline to stage the file which will then be published live * **Status:** It will be completed automatically * **Assigned to:** Correspondent LM account name. * **Suggestion:** Correction itself. It shows the current translation in the file and also the proposed translation by contributor. * **File Path:** File name with dynamic Path from GitHub.   4. Once you have provided your correspondent feedback, update the status to "Feedback Received From LM".  The status will change automatically again and the request will be assigned back to the engineering team.  **Please note**that if you do not complete the "Time Spent (min)" option, you won't be able to click on " Feedback Received From LM ".  CC_7.png  Triaging the requests  The section below explains how to handle each linguistic suggestion.  Requests that can be accepted   1. Mistranslation correction 2. Style correction 3. Terminology change request (LM should follow escalation to the terminology team) 4. New Terminology (LM should follow escalation to the terminology team) 5. Correction of source string   \*\*\*For different scenarios, escalate to the requestor.  Requests to reject   1. Requests to translate a new article/paragraph. 2. Requests to add new images or change images. 3. Transcreation requests. 4. Country specific change about feature and its specific environment. 5. If the UI string is unlocalized in content, don't accept the translated UI strings as the feedback.   \*\*\*For different scenarios, escalate to the requestor.  Requests to escalate   1. Anything with uncertainty, please send e-mail to the Moravia requestor. 2. Terminology escalation musts follow the usual processes. 3. If a change might apply to all languages, i.e the EN-US source should be updated, escalate via email to the Moravia requestor.   Implementing the changes in the files  1. Engineering Team assigns the files to the production Teams via BPI. And the will apply the changes in Oddjob.  What is GITHUB Pull Request (PR)?  Even if the LM won't be working in GITHUB, below you will find some background information on it that will help you to gain deeper understanding of this task. GITHUB Pull Request (PR) is a methodology for submitting contribution to source code repository using GITHUB.  *What are you looking for in a Pull Request?*  CC_8.png |

