# Contribute to Office 365 developer training content

Thank you for your interest in the Office 365 developer training content!

\* [Ways to contribute](#ways-to-contribute)

\* [Contribute using GitHub](#contribute-using-github)

\* [Contribute using Git](#contribute-using-git)

\* [How to use Markdown to format your topic](#how-to-use-markdown-to-format-your-topic)

\* [FAQ](#faq)

\* [More resources](#more-resources)

## Ways to contribute

Here are some ways you can contribute to this documentation:

\* To make small changes to an article, [Contribute using GitHub](#contribute-using-github).

\* To make large changes, or changes that involve code, [Contribute using Git](#contribute-using-git).

\* Report documentation bugs via GitHub Issues

\* Request new documentation at the [Office Developer Platform UserVoice](http://officespdev.uservoice.com) site.

## Before we can accept your pull request

### Minor corrections

Use GitHub to contribute to this documentation without having to clone the repo to your desktop. This is the easiest way to create a pull request in this repository. Use this method to make a minor change that doesn't involve code changes.

\*\*Note\*\* Using this method allows you to contribute to one article at a time.

### To Contribute using GitHub

### Larger submissions

1. Find the article you want to contribute to on GitHub.

If the article is in MSDN, choose the \*\*suggest and submit changes\*\* link in the \*\*Contribute to this content\*\* section and you'll be taken to the same article on GitHub.

2. Once you are on the article in GitHub, sign in to GitHub (get a free account [Join GitHub](https://github.com/join).

3. Choose the \*\*pencil icon\*\* (edit the file in your fork of this project) and make your changes in the \*\*<>Edit file\*\* window.

4. Scroll to the bottom and enter a description.

5. Choose \*\*Propose file change\*\*>\*\*Create pull request\*\*.

You now have successfully submitted a pull request. Pull requests are typically reviewed within 10 business days.

## Contribute using Git

Use Git to contribute substantive changes, such as:

\* Contributing code.

\* Contributing changes that affect meaning.

\* Contributing large changes to text.

\* Adding new topics.

### To Contribute using Git

1. If you don't have a GitHub account, set one up at [GitHub](https://github.com/join).

2. After you have an account, install Git on your computer. Follow the steps in [Setting up Git Tutorial](https://help.github.com/articles/set-up-git/).

3. To submit a pull request using Git, follow the steps in [Use GitHub, Git, and this repository](#use-github-git-and-this-repository).

4. You will be asked to sign the Contributor's License Agreement if you are:

\* A member of the Microsoft Open Technologies group.

\* A contributors who doesn't work for Microsoft.

As a community member, you must sign the Contribution License Agreement (CLA) before you can contribute large submissions to a project. You only need to complete and submit the documentation once. Carefully review the document. You may be required to have your employer sign the document.

Signing the CLA does not grant you rights to commit to the main repository, but it does mean that the Office Developer and Office Developer Content Publishing teams will be able to review and approve your contributions. You will be credited for your submissions.

Pull requests are typically reviewed within 10 business days.

## Use GitHub, Git, and this repository

\*\*Note:\*\* Most of the information in this section can be found in [GitHub Help] articles. If you're familiar with Git and GitHub, skip to the \*\*Contribute and edit content\*\* section for the particulars of the code/content flow of this repository.

### To set up your fork of the repository

1. Set up a GitHub account so you can contribute to this project. If you haven't done this already, please go to [GitHub Home] and do it now.

2. Set up your machine with Git. Follow the instructions in the [Setting up Git Tutorial].

3. Create your own fork of this repository. To do this, at the top of the page, click the \*\*Fork\*\* button.

4. Copy your fork to your local machine. To do this, go open GitBash. At the command prompt enter:

git clone https://github.com/<your user name>/<repo name>.git

Next, create a reference to the root repository by entering these commands:

cd <repo name>

git remote add upstream https://github.com/OfficeDev/<repo name>.git

git fetch upstream

Congratulations! You've now set up your repository. You won't need to repeat these steps again.

### Contribute and edit content

To make the contribution process as seamless as possible, follow these steps.

#### To contribute and edit content

1. Create a new branch.

2. Add new content or edit existing content.

3. Submit a pull request to the main repository.

4. Delete the branch.

\*\*Important\*\* Limit each branch to a single concept/article to streamline the work flow and reduce the chance of merge conflicts. Content appropriate for a new branch includes:

\* A new article.

\* Spelling and grammar edits.

\* Applying a single formatting change across a large set of articles (for example, applying a new copyright footer).

#### To create a new branch

1. Open Git Bash.

2. At the Git Bash command prompt, type `git pull upstream master:<new branch name>`. This creates a new branch locally that is copied from the latest OfficeDev master branch.

3. At the Git Bash command prompt, type `git push origin <new branch name>`. This alerts GitHub to the new branch. You should now see the new branch in your fork of the repository on GitHub.

4. At the Git Bash command prompt, type `git checkout <new branch name>` to switch to your new branch.

#### Add new content or edit existing content

You navigate to the repository on your computer by using File Explorer. The repository files are in `C:\Users\<yourusername>\<repo name>`.

To edit files, open them in an editor of your choice and modify them. To create a new file, use the editor of your choice and save the new file in the appropriate location in your local copy of the repository. While working, save your work frequently.

The files in `C:\Users\<yourusername>\<repo name>` are a working copy of the new branch that you created in your local repository. Changing anything in this folder doesn't affect the local repository until you commit a change. To commit a change to the local repository, type the following commands in GitBash:

git add .

git commit -v -a -m "<Describe the changes made in this commit>"

The `add` command adds your changes to a staging area in preparation for committing them to the repository. The period after the `add` command specifies that you want to stage all of the files that you added or modified, checking subfolders recursively. (If you don't want to commit all of the changes, you can add specific files. You can also undo a commit. For help, type `git add -help` or `git status`.)

The `commit` command applies the staged changes to the repository. The switch `-m` means you are providing the commit comment in the command line. The -v and -a switches can be omitted. The -v switch is for verbose output from the command, and -a does what you already did with the add command.

You can commit multiple times while you are doing your work, or you can commit once when you're done.

#### Submit a pull request to the main repository

When you're finished with your work and are ready to have it merged into the main repository, follow these steps.

#### To submit a pull request to the main repository

1. In the Git Bash command prompt, type `git push origin <new branch name>`. In your local repository, `origin` refers to your GitHub repository that you cloned the local repository from. This command pushes the current state of your new branch, including all commits made in the previous steps, to your GitHub fork.

2. On the GitHub site, navigate in your fork to the new branch.

3. Choose the \*\*Pull Request\*\* button at the top of the page.

4. Verify the Base branch is `OfficeDev/<repo name>@master` and the Head branch is `<your username>/<repo name>@<branch name>`.

5. Choose the \*\*Update Commit Range\*\* button.

6. Add a title to your pull request, and describe all the changes you're making.

7. Submit the pull request.

One of the site administrators will process your pull request. Your pull request will surface on the OfficeDev/<repo name> site under Issues. When the pull request is accepted, the issue will be resolved.

#### Create a new branch after merge

After a branch is successfully merged (that is, your pull request is accepted), don't continue working in that local branch. This can lead to merge conflicts if you submit another pull request. To do another update, create a new local branch from the successfully merged upstream branch, and then delete your initial local branch.

For example, if your local branch X was successfully merged into the OfficeDev/microsoft-graph-docs master branch and you want to make additional updates to the content that was merged. Create a new local branch, X2, from the OfficeDev/microsoft-graph-docs master branch. To do this, open GitBash and execute the following commands:

cd microsoft-graph-docs

git pull upstream master:X2

git push origin X2

You now have local copies (in a new local branch) of the work that you submitted in branch X. The X2 branch also contains all the work other writers have merged, so if your work depends on others' work (for example, shared images), it is available in the new branch. You can verify that your previous work (and others' work) is in the branch by checking out the new branch...

git checkout X2

...and verifying the content. (The `checkout` command updates the files in `C:\Users\<yourusername>\microsoft-graph-docs` to the current state of the X2 branch.) Once you check out the new branch, you can make updates to the content and commit them as usual. However, to avoid working in the merged branch (X) by mistake, it's best to delete it (see the following \*\*Delete a branch\*\* section).

#### Delete a branch

Once your changes are successfully merged into the main repository, delete the branch you used because you no longer need it. Any additional work should be done in a new branch.

#### To delete a branch

1. In the Git Bash command prompt, type `git checkout master`. This ensures that you aren't in the branch to be deleted (which isn't allowed).

2. Next, at the command prompt, type `git branch -d <branch name>`. This deletes the branch on your computer only if it has been successfully merged to the upstream repository. (You can override this behavior with the `D` flag, but first be sure you want to do this.)

3. Finally, type `git push origin :<branch name>` at the command prompt (a space before the colon and no space after it). This will delete the branch on your github fork.

Congratulations, you have successfully contributed to the project!

## How to use Markdown to format your topic

### Standard Markdown

All of the articles in this repository use Markdown. While a complete introduction (and listing of all the syntax) can be found at [Markdown Home], we'll cover the basics you'll need.

If you're looking for a good editor, try [Markdown Pad].

### How do I get a GitHub account?

Fill out the form at [Join GitHub](https://github.com/join) to open a free GitHub account.

### Where do I get a Contributor's License Agreement?

You will automatically be sent a notice that you need to sign the Contributor's License Agreement (CLA) if your pull request requires one.

As a community member, \*\*you must sign the Contribution License Agreement (CLA) before you can contribute large submissions to this project\*\*. You only need complete and submit the documentation once. Carefully review the document. You may be required to have your employer sign the document.

### What happens with my contributions?

When you submit your changes, via a pull request, our team will be notified and will review your pull request. You will receive notifications about your pull request from GitHub; you may also be notified by someone from our team if we need more information. If your pull request is approved, we'll update the documentation. We reserve the right to edit your submission for legal, style, clarity, or other issues.

### Can I become an approver for this repository's GitHub pull requests?

Currently, we are not allowing external contributors to approve pull requests in this repository.

### How soon will I get a response about my change request?

Pull requests are typically reviewed within 10 business days.

## More resources

\* To learn more about Markdown, go to the Git creator's site [Daring Fireball].

\* To learn more about using Git and GitHub, first check out the [GitHub Help].

\* [Microsoft Contributor License Agreement](https://cla.opensource.microsoft.com/)

[GitHub Home]: http://github.com

[GitHub Help]: http://help.github.com/

[Setting up Git Tutorial]: http://help.github.com/win-set-up-git/

[Markdown Home]: http://daringfireball.net/projects/markdown/

[Daring Fireball]: http://daringfireball.net/