VERSION AND RELEASE MANAGEMENT

The latest version of magento is 2.4.5-p1.

Before the release notes of this version they have versions

- 2.4.5
- 2.4.4-p2
- 2.4.4-p1
- 2.4.4
- 2.4.3-p3
- 2.4.3-p2
- 2.4.3-p1
- 2.4.3
- 2.4.2-p2
- 2.4.2
- 2.4.1
- 2.4.0 etc.

The 2.4.5-p1 version is a security release that provides five security fixes that enhance your Adobe Commerce 2.4.5 or Magento Open Source 2.4.5 deployment. It provides fixes for vulnerabilities that have been identified in the previous release (Adobe Commerce 2.4.5 and Magento Open Source 2.4.5).

This security patch includes five security bug fixes.

- One fix included the creation of a new configuration setting.
- The Require email confirmation if email has been changed configuration setting lets administrators require email confirmation when an admin user changes their email address

ISSUE LIFECYCLE MANAGEMENT

The GitHub issue workflow ensures that issues are clear, they'll written, and thoroughly vetted.

- Reporter - The user who filed the initial issue report.
- Maintainer - A member of the Community Maintainers Team who is working on the issue report to update and confirm the report in accordance with all requirements.
- Automated Contributor Assistant - Non-human users/bots that perform automatic checks and provide assistance to human users.
- Label - The GitHub label applied to the ticket.

Following these procedures allows valid issues to get the attention they deserve.

Issues that are reported on the public github must pass through a series of gates, or stages of quality assessment, to ensure that their quality meets their standards. There are three gates, and an issue must pass through all three of these assessments before they transfer it to either core developers or community developers. The purpose of these gates is to identify core technical issues common to all reported tickets, and to show the progress on each reported issue.

Issue Gates

Issue gates are a series of steps that are run to make sure the issue has all the information needed to reproduce the bug. This includes system configurations, required configurations, reproduction steps and any other required information.

- Gate 1: Verification of the report format - ensures that report content and structure meet all of their requirements.
- Gate 2: Manual verification - Someone manually confirms that all necessary information has been provided: steps to reproduce, system configs, etc.
- Gate 3: Reproduce the bug - Someone sets up an environment and tries to reproduce the bug. The issue is then confirmed or closed.

Gate 1 - Verification of the report format

The main goal of this initial verification stage is to be sure that the report has well-structured content that meets the requirements from issue reporting guidelines.This stage looks trivial and formal, but it will definitely affect

processing speed. Generally, an issue that has the expected structure and clear information will be processed faster than the same report with a poor format. The reported issue must contain all the following keywords in the description section:

- Preconditions
- Steps to Reproduce
- Actual Result
- Expected Result

A maintainer can request the reporter to update the issue description and provide additional information. The maintainer will add the Issue : needs update label. A reporter has 14 days to update the issue description; otherwise, the issue will be closed. The maintainer may update the issue description format if sufficient information is provided.

Gate 2 - Manual verification

Gate 2 verifies a submitted issue is ready for development. By the end of the process, it has been vetted for development including all labels for Functional Areas, Affected Versions, and so on. The reproduction steps are correct and the reported issue is a defect that should be fixed and not due to misuse or a misconfiguration.

Working on an issue report as a reporter, maintainer, or developer is always a commitment. It is beneficial for every participating party to monitor activity and comments on the ticket during its lifetime, and provide any needed information or insights.

Preparation

These are steps for reviewing the issue, verifying reproduction steps, and assigning a maintainer to work it.

1. A maintainer selects an unprocessed ticket from the GitHub tracker. The recommended tool is the issue confirmation and triage board.
2. The maintainer reviews the list from the “Ready for Confirmation” column and selects an issue to begin processing.
3. After selecting the ticket, the maintainer checks the description and reproduction steps.

4. When the maintainer is ready to start processing the issue, the maintainer should assign the ticket to him/her self. This indicates someone is actively working on the issue.

Validation

These are the steps for validating the issue format and all information provided checks out:

1. When the issue is entered, verify that it meets all requirements from the templates and issue reporting guidelines.If the format is not valid, the maintainer should read the report carefully and edit the issue to better match one of the required templates.
2. The maintainer can select the issue and review all information, reproduction steps, etc. If the information is incomplete, the maintainer requests more information from the reporter and applies the label Issue : needs update. All work pauses on this ticket until the reporter provides more information.
3. If the ticket has enough information, the maintainer analyzes the problem described in the ticket: described steps to reproduce are valid, expected behaviour is valid, the configuration described in preconditions is valid.
4. Is it validated?

- If all provided information is clear and sufficient, it is validated.
- If it is not validated, the maintainer adds the label Issue : needs update and requests more information from the reporter.

Finalization

Steps for final review of an issue for contributors/developers to work on the issue.

1. Please make sure that all required conditions are met:
    - Issue format is valid.
    - Issue is reproducible with one of the supported versions and labelled appropriately.
    - Area: XXX is label applied to the ticket.
    - (optional if possible) Reported on : XXX label is applied to the ticket.
    - (optional if possible) Severity: XX label is applied to the ticket.
    - Add the label Issue: Confirmed to the ticket.

2. Wait for a response from the Automated Contributor Assistant, which normally takes 30-60 seconds.
3. If all required information was provided, the Automated Contributor Assistant will comment with reference ticket numbers and a link in the internal backlog. Otherwise, the label Issue: Confirmed will be removed, and information on what is missing in the report will be provided to the maintainer.
4. Un-assign the ticket from yourself so that developers can claim the issue and start development.

If the issue was reproduced on gate 3, they will create an internal AC-XXXX ticket to track the progress of the issue.

Once an issue has been acknowledged and confirmed, it goes through the Triage Process and is prioritised. After triage, either core developers or community developers may fix it. they encourage everyone to join the Community Contribution Team and submit Pull Requests with the bug fix to magento/magento2 repository.

OPEN SOURCE CONTRIBUTION REQUIREMENTS FOR DEVELOPERS

Contribute to Magento DevDocs

Share your knowledge with the community by contributing to Magento DevDocs! You can contribute by creating an issue or pull request (PR) on the DevDocs GitHub repository. Magento welcomes all types of contributions; from minor typo fixes to new topics.

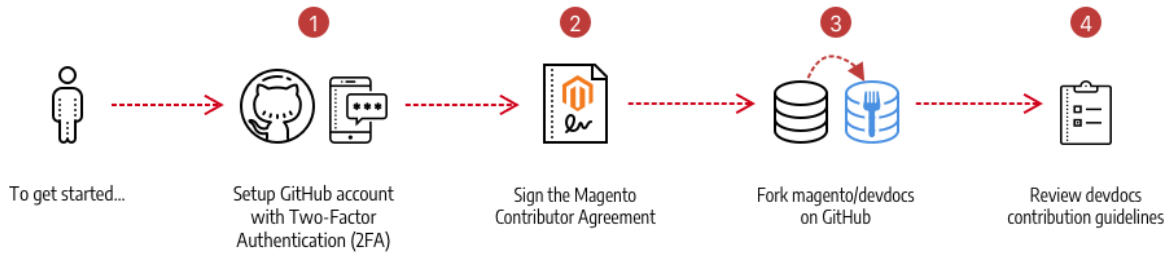
DevDocs staff members and Community Maintainers review issues and pull requests on a regular basis. Magento does their best to address all issues as soon as possible, but working through the backlog takes time. Magento appreciates your patience.

Rewards for contributions

If you write and contribute a full topic, they will add your name (or your company's name) at the top of the DevDocs page and link it to your blog or website.

If you contribute a new topic or a major update to a topic, your GitHub username will be added to a description of the update in their What's New topic.

Get started



1. Make sure you have a GitHub account.
   Note for partners: Add 2FA protection when contributing to Magento repositories.
2. Fork the DevDocs repository. Remember to sync your fork and update branches as needed.
3. Review the DevDocs guidelines.

Note: If you use a fork instead of a branch, please set permissions to allow maintainers to edit and update the PR. See Allowing changes to a pull request branch created from a fork in the GitHub documentation.

Contribute documentation

The following diagram shows the contribution workflow:

Create a branch

1. Create a new branch from your fork using a name that best describes the work or references a GitHub issue number.
2. Edit or create markdown (.md) files in your branch.
3. Push your branch to your fork.

Create a pull request

1. Create a pull request to the magento/devdocs repository.
   In general, you should use master as the base branch when creating a PR. If your contribution is related to a release that is in progress, use a version-specific integration branch, such as develop.
2. Complete the pull request template. Review the Pull Request Process page to learn how to complete a PR (with info about completing the template, adding a whatsnew , and more.)
   they will close your pull request if you do not complete the template.
3. After creating a pull request, a DevDocs staff member or maintainer will review it and may ask you to make revisions.
   they will close your pull request if you do not respond to feedback in two weeks.

Contribution guidelines

The following guidelines may answer most of your questions and help you get started:

- Write content using Markdown.
- Review existing pull requests and issues to avoid duplicating work.
- For large contributions, or changes that include multiple files, open an issue and discuss it with us first. This helps prevent duplicate or unnecessary work.
- Do not make global find-and-replace changes without first creating an issue and discussing it with us. Global changes can have unintended consequences.
- Do not make changes to content in the _data/codebase directory, which contains auto-generated data from Magento source code. Any manual changes will be lost when the file regenerates.
- Combine multiple small changes (such as minor editorial and technical changes) into a single pull request. This helps us efficiently and effectively facilitate your contribution.

- Familiarise yourself with the organisation and conventions of their existing documentation before creating a pull request. Changes that are consistent with their style and conventions have a higher acceptance rate.
    - If you need to update the site navigation, ask for help in Slack (#devdocs).
- Ensure that you update the correct version(s) of documentation (v2.3). If you are not sure what directory to put your content in, just do your best. they can help relocate it (if necessary) during the review process.
- Review your work for basic typos, formatting errors, or ambiguous sentences before opening a pull request.
- Revise pull requests according to review feedback. They will close pull requests that require an inordinate amount of time to review and process (especially for minor changes) if you fail to make revisions according to review feedback.
- Do not directly contact DevDocs team members or maintainers on Slack to review your pull request unless it has been open for more than five days. They have a process and queue for pull requests that everyone must follow.
- Get recognized on the DevDocs web site for writing new topics! Add your name and a link to your company website or GitHub profile to the file metadata so that they can display it on the page.
- They no longer recognize individual community members who contribute features to the Magento 2 codebase in the corresponding feature topic(s) on the DevDocs website. Magento recognizes these contributions in more appropriate channels (for example, the Magento DevBlog).

Tips for writing content

Use the following guidelines to help you with the writing process:

- Focus your efforts on providing useful information for your fellow Magento developers and community members. For example, consider providing or revising code samples, important notes, and clarifying vague or ambiguous content.
- Define the goal of your topic. What exactly do you want to teach the reader?
- Make the title of your topic reflect the content.
- Keep your sentences concise. Separate conceptual information from procedural steps.

- Batch several small changes into a single pull request (instead of separate ones) to ensure your contributions are approved and merged quickly. Have several typo fixes across several areas of documentation? Combine them into a single PR.
- Remember to write in present tense, use the second person, and use active voice (not passive). For example, ″The log captures commands, output″.
- Use notes to alert readers about important details.
- Use cross-references to other topics sparingly. they can help you with the syntax if it is not clear.

Templates

they provide templates to help you get started writing new content and understanding Markdown formatting:

- General topic template - Markdown | HTML: This is a template for writing any topic with example formats and styles.
- Tutorial templates: These templates provide example formats and styles for step-by-step instructions (like how-tos). Each file adds navigation buttons when content is generated. Templates include:
    - First introduction topic - Markdown | HTML: Introduction to a tutorial for prerequisites and listing steps
    - Middle topic - Markdown | HTML: Use for each step in a tutorial.
    - Final step topic - Markdown | HTML: Use for the last step of the tutorial.

Edit metadata

The Markdown (.md) file's metadata is a set of YAML key-value pairs. The metadata section is located at the top of each file.

`group:`

`title:`

`contributor_name:`

`contributor_link:`

Key-value pair reference:

Property	Description
group	Defines the topic's guide or section. Use the table of contents `.yml` file name. This loads your left-side navigation. they will help during the PR process to add new files to the `.yml` file.
title	Sets the title of the page in the HTML metadata and the main title on the page.
contributor_name	Sets the name of the contributor who wrote the topic and displays it on the page.
contributor_link	Creates a link to the contributor's GitHub profile or company website.

Report an issue

If you find a typo or errors in Magento DevDocs, you can either fix it with a pull request (as described above) or you can report it by creating an issue in the DevDocs GitHub repository.

You must complete the issue template. they will close your issue if you fail to complete the template. Enter as much information as you can, including content corrections, steps to reproduce, command or code updates, or questions for clarifications.

REVIEW PROCESS

They will review your extension code and submit information according to the queue, sending feedback and confirmation by email. Track the status and progress of your extension submission through your Marketplace account. Their Extension Quality Program (EQP) verifies that all Marketplace extensions meet Magento quality standards and best practices.

Before submitting your extension for review, complete the information about the extension through the developer portal.
The extension submission process requires you to do the following:

- Review and agree to the Terms and Conditions of Commerce Marketplace.
- Pass both technical and marketing review.
- Provide information about your business to ensure that transactions and payments are processed efficiently. This information includes W-8 / W-9 forms, as required by law. For more information, see the technical and marketing review guidelines.

You can submit your extension for technical review when you are ready to have Commerce Marketplace review the code and technical aspects of your extension. These steps require additional information for your extension including a PDF guide, supported Magento versions, and code package.

You can submit your extension for marketing review as soon as your marketing content is ready for review.

You must complete both the technical and marketing review to fully list your extension on the Commerce Marketplace.

Submit for Technical Review

All extensions submitted to Commerce Marketplace must pass the automated technical review as part of the extension submission workflow. Technical review helps to improve the quality of products on Commerce Marketplace by checking for indications of plagiarism, malware, and adherence to Magento coding standards. Developers whose extensions do not pass technical review receive a report of the results. After the issues are resolved, you are welcome to resubmit the extension. Extensions must pass technical review to receive a listing on Commerce Marketplace.

When your extension entry is complete, you can submit your extension for technical review. During the process, we review the code according to technical guidelines, install and use the extension according to your documentation, and verify specifics from your submission form. You can track

the status and progress of your extension submission through your Marketplace account.

Prepare for Technical Review

Before submitting an extension or theme for marketing review, conduct your own internal review of the content to make sure that it is ready for publication.

- Review the technical guidelines to ensure that your extension meets Commerce Marketplace and Magento development requirements.
- For theme extensions, verify all image, css, and code assets correctly load on the storefront. For feature / service extensions, make sure data and options follow coding standards, logging, etc.
- All extensions must be secure, without viruses, malware, or vulnerabilities.
- Fully test your extension, including installation, dependencies, shared packages, configuration, and usage.

You will receive email confirmation when the extension is submitted for Technical Review, and will be notified when the review is complete.

USER MANUAL

https://docs.magento.com/user-guide/

INSTALLATION

https://experienceleague.adobe.com/docs/commerce-operations/installation-guide/overview.html

TUTORIALS

https://experienceleague.adobe.com/docs/commerce-learn/tutorials/overview.html