



WHITE PAPER

Adobe Commerce 2.4 Upgrade Guide



Table of Contents

Introduction	4
Who is this guide for?	4
Overview	5
2022 Release Calendar	6
Adobe Commerce upgrade journey	7
Project Launch	7
Annual Planning	8
Upgrade execution	8
Upgrade Analysis	8
Development & QA	9
UAT & Preparing to Launch	10
Launch	10
Post-Launch	10
Preparing to upgrade to Adobe Commerce 2.4.4	11
Platform changes in 2.4.4	11
1. Start working on upgrading NOW!	11
2. Make sure your system is ready	11
System Requirements (as of 2.4.4)	11
Prerequisites	12



3. Understand the scope of upgrading	13
Backward incompatible changes	13
Third-party extensions	13
Custom Modules	13
[HOT TIP] The Upgrade Compatibility Tool	14
4. Implementing the upgrade	16
Upgrade Best Practices	18
Adobe Commerce Recommended 2022 Upgrade Paths	20
Upgrading from versions 2.3.0-2.3.6	20
Upgrading from 2.3.7	22
Upgrading from 2.4.0-2.4.2	24
Upgrading from 2.4.3	24
Upgrade Plan Checklist	25





Introduction

Upgrading your Adobe Commerce platform is critical to ensuring that your eCommerce store stays secure, performant and PCI-compliant. The process can seem complex, but we've developed the following guide to walk you through the key considerations when preparing for an upgrade.

The guide is broken into three components. First, we provide an overview of the typical Adobe Commerce upgrade journey and best practices to follow along that journey. Then we take a deep dive into the upgrade process with a timely example and provide step-by-step instructions for upgrading to Adobe Commerce version 2.4.4. This upcoming patch release will be generally available on March 8, 2022, and it's important to start preparing for this upgrade early as the [End-of-Life \(EOL\)](#) dates are approaching for both the 2.3 line and versions 2.4.0-2.4.3 in 2022. Lastly, we'll provide planning resources and upgrade tools that will make your upgrade process more seamless and efficient.

To ask any questions about upgrades or to provide feedback to this guide, join our [upgrade community on Slack!](#)

#upgrading-adobe-commerce

Who is this guide for?

eCommerce Managers and Technical Directors

This guide will help these roles understand the Adobe Commerce upgrade journey, the importance of upgrading regularly, and how to best plan and prepare for an upgrade.

Operations and Development Teams

This guide will help these teams learn the technical steps needed to upgrade to 2.4.4 (or any Adobe Commerce version) and the tools they can use to make the process easier, faster and more affordable.

Overview

One of the reasons you chose Adobe Commerce likely includes its broad out-of-the-box feature set, its SaaS features offered separate from the core code, its robust offering of Marketplace extensions, or its unique ability to allow for infinite flexibility so that you can customize your site to best serve your business' and customers' needs. However, the benefit of being a highly extensible and customizable product can spur potential upgrade issues when customizations are not coded to best practices, leading to higher-than-expected upgrade costs.

So... why upgrade at all?

Upgrading empowers your business to stay nimble in the fast-paced and ever-changing eCommerce industry and enables your platform to be always-compatible with the latest features that help maximize sales and conversions. Including upgrades in your regular maintenance plans is also critical to ensuring that your store stays secure, performant and PCI-compliant.

Security is one of the top reasons for upgrading as 83% of security incidents occur on outdated software. According to [IBM](#), the average cost of a data breach is \$3.86 million – far greater than what it costs to mitigate this risk through upgrading. Adobe Commerce offers two ways to keep your store secure throughout the year: patch releases and security patch releases. Patch releases include security, performance, and high-priority bug fixes. Security patches include fixes and enhancements to keep your site secure and are easier to implement.

Performance is another top reason for upgrading. According to [HubSpot](#), the first five seconds of load time have a significant effect on conversion rates and every second of latency thereafter has a -4.4% impact. That, coupled with the fact that page speed is a leading SEO ranking factor, demonstrates why site performance is a critical element of your site to maintain and regularly improve upon. Each Adobe Commerce patch release includes performance improvements, so taking advantage of new releases will support your growth plans and keep your business competitive.

The case for delaying or deferring platform upgrades often comes down to the immediate cost. However, the real cost of running an outdated version of any software is much larger and can have a lasting impact on a business. It may seem counterintuitive, but performing regular platform updates requires less overall effort

than performing infrequent updates due to the amount of accumulated technical debt that results from delaying.

Adobe Commerce recently worked with a partner who has a retail merchant that used to conduct upgrades infrequently and inconsistently (annually or longer). By transforming how they approach upgrades and following an Adobe-recommended regular upgrade path over the course of 12 months, the partner was able to save the client four weeks' worth of cumulative development time, effort and associated costs, all of which were redirected to initiatives that drive business growth.

83% of security incidents occur on outdated software. According to IBM, the average cost of a data breach is **\$3.86M**



When updates are performed regularly, changes are incremental and the corresponding upgrade effort reflects this. When platform updates are deferred for an extended period of time, they can become a much more involved process. Furthermore, the extensions you use from the [Marketplace](#) as well as any other third-party integrations may also be impacted. Lastly, the time it takes to investigate, plan, and perform a delayed upgrade are all extended, which adds avoidable effort and costs.

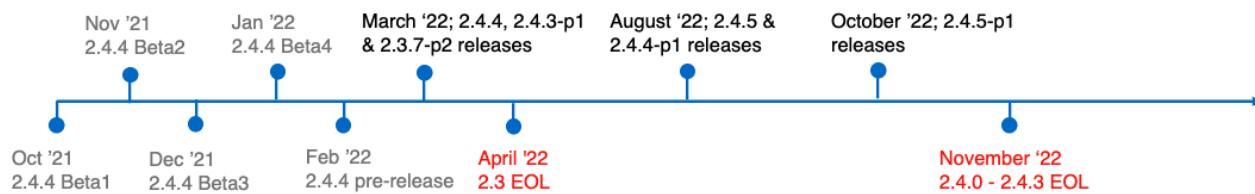
Continued growth in digital commerce has applied increased pressure on businesses to evolve faster, more often, and in unpredictable ways. Failure to keep up and anticipate customer purchasing behavior has leveled the playing field or even the biggest, most established brands. You must be able to provide robust, personalized experiences across all touchpoints, with no lapses in performance and uptime. You must be able to innovate faster, without limits, to stay ahead of global competitors. By upgrading, you're future proofing your business and setting yourself up to better service dynamic customer needs.

2022 Release Calendar

Adobe Commerce publishes a [release calendar](#) annually to facilitate merchants' planning processes and recommends upgrading each patch release cycle. To remain PCI-compliant, merchants must be on the latest patch or security patch. The following timeline shows the major release and EOL events in 2022.

Important events to note include:

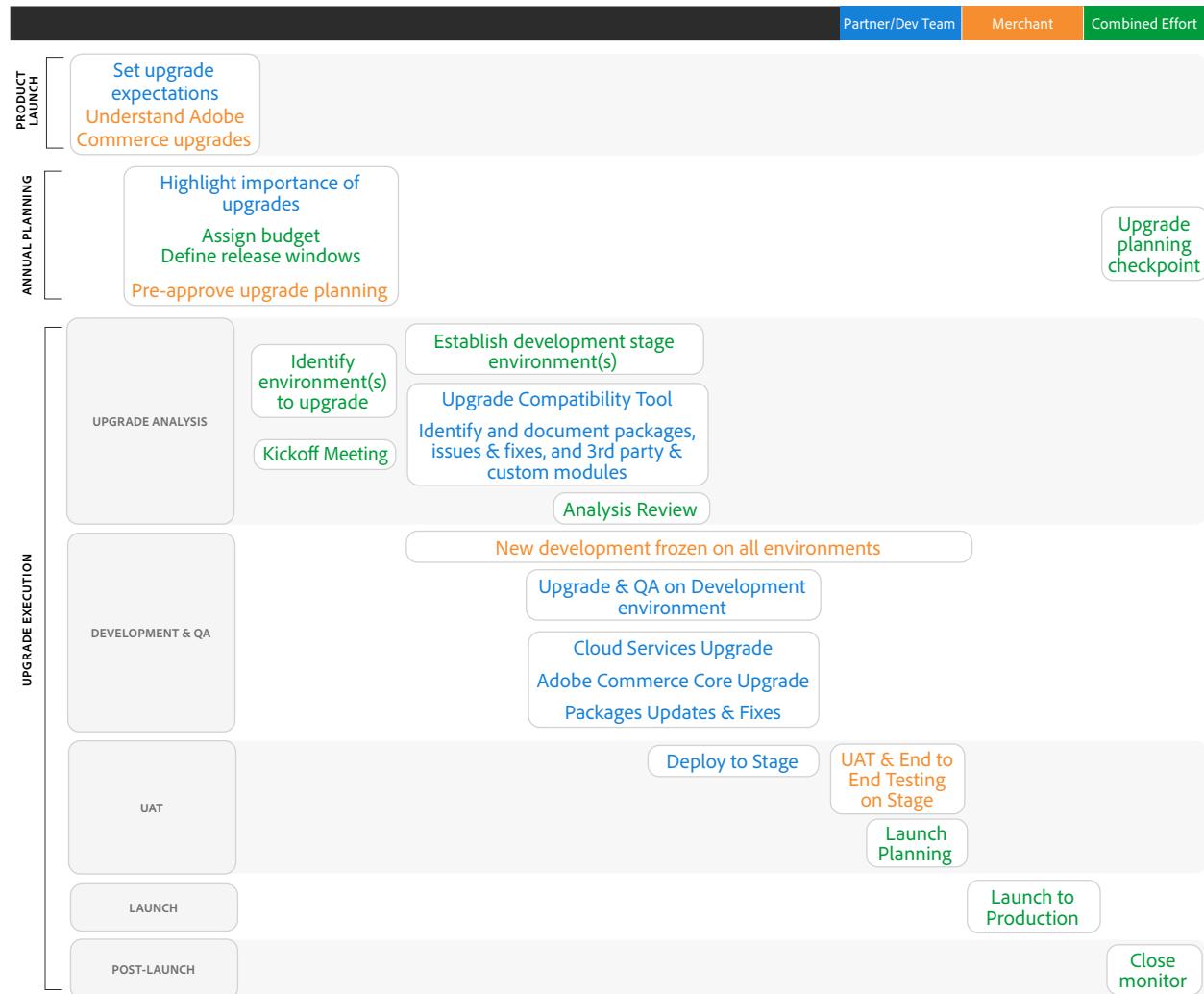
2022 Release Timeline



- 2.3 line support reaches EOL in April '22
- 2.4.0 - 2.4.3 (based on PHP 7.4) will reach end of quality support in November '22, when PHP 7.4 reaches EOL
- Based on these two EOL events, it is important to upgrade to version 2.4.4 or higher by November '22
- In line with the [Adobe Commerce lifecycle policy](#), versions 2.4.4 and 2.4.5 will receive quality support until November '24

Adobe Commerce upgrade journey

Upgrades require close attention, planning, and management by both the merchant and partner/development team. To help you understand the Adobe Commerce upgrade journey, we will describe the process in three main phases: product launch, annual planning, and upgrade execution.



Project Launch

Maintaining a healthy eCommerce site is just as critical to the success of your business as your initial launch, and a key aspect of that is proactively planning and preparing for upgrades that will help your site operate as smoothly as possible. Not setting the right upgrade expectations during this time will assuredly lead to future dissatisfaction. Make sure you are aware of and understand the [Adobe Commerce release cadence](#) and use this guide to help you define how and when to plan for upgrades.

Annual Planning

Many companies work from annual budgets and roadmaps. It is imperative during these annual discussions that you talk about your platform health, direction, and **upgrade strategy** for the year, along with how it fits into the overall goals and KPIs of the business. Following the [Adobe Commerce release calendar](#), assign budget and release windows for upgrades, avoiding holidays, company closures and peak times. By understanding the Adobe Commerce release calendar and staying informed of what changes and improvements are part of each release through our [release notes](#), you can better plan your year.

During planning, you may want to ask yourself:

- Are there any customizations or extensions that need to be built to meet the goals of the business?
- Am I checking that these customizations/extensions are not provided natively by Adobe Commerce and will not be in an upcoming release? If they are, I might be able to save the company both install time and money
- Do I have any third-party integrations that may need maintenance during the upcoming year?

To ensure your team or merchant's team is on the right track, plan for an upgrade checkpoint midway through the year to review plans, budget, timeline, and capacity.

Upgrade execution

The upgrade execution phase consists of five phases – upgrade analysis, development & QA, UAT & Preparing to Launch, Launch, and Post-Launch.

Upgrade Analysis

This is arguably the most important part of the upgrade process. A well-executed analysis will save you time and limit surprises in the future. The outcome of this phase should be a detailed upgrade checklist and document with all dependencies.

The following are items that you may want to include in a thorough analysis:

- **Scope of target release.** Documentation on [DevDocs](#) and information from [partner release webinars](#) (available to partners only) will provide all the details you will need to know about your target upgrade
- **Upgrade Compatibility Tool results.** This tool makes any upgrade faster and easier by comparing your current code to the target version's code and producing a report of all the issues that need to be addressed. See the [Upgrade Compatibility Tool section](#) on this guide for more details. Key details to keep track of from the report include:
 - Current installed version
 - Target upgrade version
 - Number and details of critical errors found



- **Upgrading services to support target version.** Use this table template to map out which services you will need to upgrade.

Service	Current Version	Upgrade Version	Notes
PHP	7.2.33	7.4	
Redis	5.0.5	6.0	
Rabbitmq	3.7	3.8	Not currently being used but we should consider using
MariaDB	10.2.33	10.4	
Composer	1.9.2	2.0	
Elasticsearch		7.9	

- **Extensions and third-party modules.** Use this table template to help you understand the status of your extensions and customizations, so that you can make strategic decisions and define actions. This is an opportunity to replace any extensions that might be native to Adobe Commerce to minimize your project's complexity.

#	Extension / Module Name	Composer package	Vendor	Current version	Functionality	Status	Compatible with latest AC version?	Issues	Native in AC?	Action	Notes
1	Extension name and link	extension/extension-magento-2	Vendor name	Version installed	Business requirements	Enabled / Disabled	Yes/No	List identified issues faced with this extension	Yes/No	Keep/Replace/Remove	

- **Custom modules.** Similar to the third-party modules table, this template will help you track and understand the status and actions required for upgrading custom modules.

#	Module	Functionality	Required	Native in AC?	Action	Notes
1	Module name	Business requirements	Yes/No	Yes/No	Keep/Replace/Remove	

- **Composer packages and dependencies in composer.json that require an update.**

In addition, partners can participate in the [Adobe Commerce beta program](#) and use pre-release opportunities to get early access to the code for an upcoming release. Getting access to the code early helps developers prepare with enough time to complete the upgrade by the Generally Available (GA) date. Beta code is typically released five weeks in advance of the GA date and pre-releases are released two weeks in advance. For version 2.4.4, Adobe Commerce started releasing beta code five months in advance of the GA date (March 8, 2022), so partners can start preparing for that upgrade now by [signing up for the program](#).

Development & QA

Testing is the phase of an upgrade that requires the most time. As a result, this process should be as automated as possible. The [Application Testing Guide](#) provides details on how to set up and use platform and system testing tools for faster QA. Use a staging environment to test and validate your upgrade before moving to production.

UAT & Preparing to Launch

UAT is one of the last stages of the upgrade that requires reviewing and validating the site. You must also decide when to deploy and whether you need a maintenance page. Make plans for cron processes and third-party messages.

As the deployment date draws near, communication is essential. If more people know about the change on the horizon, how it will impact them, and how they need to address it, then you are more likely to have a successful launch. Do not be afraid to over-communicate every step of the way – it increases the likelihood of glowing reviews from everyone involved once you go live!

Launch

Complete the upgrade by deploying to production and updating extensions. Make sure you test critical path flows with simulated orders. Check out our [best practices section](#) within this guide for some tips on launching with minimal issues.

Follow your communication plan and ensure all stakeholders are aware of the upgrade and are fully trained to support it. Finally, debrief with your team to determine lessons learned and pitfalls. This will help you to improve the process next time.

Post-Launch

After your site launches, make sure to check your analytics data, Google Search Console, and other resources to ensure that there are no unexpected issues and that everything is working as expected.

It is always a good idea to keep an eye on performance through well-designed monitoring tools. There are many tools and means of monitoring your site performance, so be sure to pick one that pairs well with your organization. We recommend that Adobe Commerce customers who utilize our cloud infrastructure management system take advantage of services such as [New Relic](#) to monitor site performance.

Preparing to upgrade to Adobe Commerce 2.4.4

Platform changes in 2.4.4

Adobe Commerce 2.4.4 will be generally available on March 8, 2022 and [beta code](#) is now available for partners to start preparing. The 2.4.4 release is unique because it will include many platform updates. It will be powered by PHP 8.1 together with several key framework upgrades including:

- Removing support for PHP 7.3
- Support for new OpenSearch Search & Analytics Suite (Elasticsearch Parallel)
- Latest version support of multiple system dependencies including Elasticsearch, Redis, MySQL, JS libraries, and composer dependencies
- Removal of older JS libraries, redundant Laminas libraries, and NPM dependencies

Learn more about the technical platform changes on the [Adobe Commerce Devblog](#).

1. Start working on upgrading NOW!

From October '21 to February '22, partners can [subscribe to the Adobe Commerce beta program](#) to prepare for 2.4.4 GA (March 8, 2022). There will be four beta releases leading up to GA date.

- **2.4.4 Beta1:** October 18th, 2021
- **2.4.4 Beta2:** November 22nd, 2021
- **2.4.4 Beta3:** December 14th, 2021
- **2.4.4 Beta4:** January 10th, 2022
- **2.4.4 Pre-Release:** February 22nd, 2022
- **2.4.4 General Availability:** March 8th, 2022

2. Make sure your system is ready

It's important to understand what is necessary to run Adobe Commerce or Magento Open Source. You must review the prerequisites and system requirements for the version you are planning to upgrade to. All details are available on [Adobe Commerce DevDocs](#).

System Requirements (as of 2.4.4)

These software versions were officially tested and supported. Newer versions of required software may work but have not been tested, and are therefore, not officially supported. Adobe Commerce system requirements can be found on [Adobe Commerce DevDocs](#).

- Composer 2.x
- OpenSearch 1.1 or Elasticsearch 7.15



- MariaDB 10.4 or MySQL 8.0
- PHP 8.1
- RabbitMQ 3.8
- Redis 6.2
- Varnish 7.0
- Nginx 1.8 or Apache 2.4



Prerequisites

These are the instructions you must complete before you upgrade your system. See the [Update & Upgrade Checklist](#) for full details.

1. Ensure all software is up to date

See the [System Requirements](#) and [Prerequisites](#) guides for full instructions. Be sure to check out the appropriate PHP migration guides as well for compatibility issues. See the [PHP 7.4](#), [PHP 8.0](#) and [PHP 8.1](#) guides for more info. Check out the PHP system requirements in these [DevDocs](#).

2. Verify Elasticsearch is installed

Adobe Commerce 2.4 requires Elasticsearch to be installed in order to use the software. Check the [catalog search engine](#) for full details.

From the command line, run the following:

```
bin/magento config:show catalog/search/engine
```

It should return a value of `elasticsearch7` to be compatible with 2.4.4.

3. Set the open files limit

Setting the open files limit (`ulimit`) can help avoid failure from multiple recursive calls of long query strings or issues with using the `magento setup:rollback` command. This command is different for different UNIX shells. Consult your individual flavor for specifics of the `ulimit` command.

After switching to the [file system owner](#), enter the command:

```
ulimit -s 65536
```

4. Verify cron jobs are running

Magento requires 3 cron jobs, all running as the file system owner. See [Set Up Cron](#) for full details. Run the following:

```
crontab -l
```

It should output results similar to the following:

```
* * * * * /usr/bin/php /var/www/magento2/bin/magento
cron:run ...
* * * * * /usr/bin/php /var/www/magento2/update/cron.
php ...
* * * * * /usr/bin/php /var/www/magento2/bin/magento
setup:cron:run ...
```

5. Set DATA_CONVERTER_BATCH_SIZE

Adobe Commerce 2.4 includes security enhancements that require some data to be converted from serialized to JSON. This occurs during the upgrade, so you should increase the size to ensure a smooth upgrade. This setting requires memory, so test this value before applying your upgrade, especially if you have large amounts of data in your database.

Enter the following from the command line:

```
export DATA_CONVERTER_BATCH_SIZE 100000
```

6. Verify system permissions

File system ownership is vital for security. You must verify that the proper file owners are set with appropriate read/write/execute permissions. See [Ownership & Permissions](#) for full instructions.

3. Understand the scope of upgrading

Identify and document packages, issues and fixes, and third-party and custom modules. In these [release notes](#), you will find details on the scope of the new version.

Backward incompatible changes

As part of the 2.4.4 release, Adobe has updated multiple PHP and JavaScript dependencies and libraries. All backward incompatible changes for 2.4.4 beta versions are described in these [DevDocs](#). Please note, before the release GA date, you must have partner access and be subscribed to the beta program to have access to these pages. All users will have access after the GA date.

Third-party extensions

Adobe Commerce Marketplace's [new compatibility policy](#) will ensure that ALL extensions listed are compatible with the latest released version within 30 days of the GA date. For this reason, it is important to get your third-party extensions, whenever possible, through the [Marketplace](#).

Custom Modules

All custom modules should be checked against the target version you are looking to upgrade to. This is the most time- and resource- intensive process of an upgrade. When evaluating your custom modules, you must



look for backward incompatible changes and be aware of new practices, such as Controller decomposition . You can find more details within [release notes](#). Additionally, ensure that you're following [best practices for module development](#).

[HOT TIP] The Upgrade Compatibility Tool

The [Upgrade Compatibility Tool](#) is a command-based tool that analyzes your instance for potential upgrade issues. It checks for issues between the current version you have installed and the version you are trying to upgrade to. Using this tool will reduce the effort required from your team to understand the scope and impact of the upgrade. It'll help you avoid common code issues when upgrading and provides clear direction on how to resolve issues found. It also helps prioritize the most critical issues necessary to ensure a successful upgrade, saving both time and costs when upgrading.

1. Download The Tool

[Download & Install the tool](#) via composer. It requires PHP 7.3 or later, at least 2GB of RAM, Node.js if you are checking GraphQL compatibility, and an Adobe Commerce license.

```
composer create-project magento/upgrade-compatibility-tool uct --repository https://repo.magento.com
```

2. Run The Tool

The main command to use is `upgrade:check`. It analyzes your instance and checks for errors, warnings, and critical issues in your instance.

Run the command:

```
bin/uct upgrade:check <dir> -c <coming version>
```

`<dir>` is the directory of your code base.

`-c` is the version you want to check against (example: 2.4.4).

Some more options to use with this command are:

```
--ignore-current-version-compatibility-issues
```

This suppresses all known critical issues, errors and warnings against your current Adobe Commerce version. It will only provide errors against the version you are trying to upgrade to.

```
--min-issue-level
```

This setting allows you to set the minimum issue level, to help prioritize only the most important issues with your upgrade. The options are warning, error, and critical in ascending order of severity.

```
-m | [=MODULE-PATH]
```



If you want to analyze only a certain vendor, module, or even directory, you can specify the path as an option as well.

```
--vanilla-dir
```

This allows you to check core code for any nonstandard implementation of features or customizations. It is important that these are cleaned up beforehand. A vanilla instance of your version will be automatically downloaded for reference. (Note: this can also be done with the `core : code : changes` command in the tool). Using these commands together help you identify the most critical issues for your team to address.

Run the command:

```
bin/uct upgrade:check /path/to/magento/ --ignore-current-compatibility-issues -min-issue-level critical  
--vanilla-dir /path/to/vanilla/code/ /path/to/magento/app/code/Vendor/
```

3. Analyze The Output

The Upgrade Compatibility Tool exports a `json` file report identifying the affected code or modules, severity, and a description of the problem for every issue encountered. It will also output a summary report with a complexity score, which will allow your teams to understand at a rough order of magnitude what it will take to upgrade to the latest version. The lower the complexity score, the easier it will be to perform the upgrade.

Installed version	2.3.5-p2
Adobe Commerce version	2.4.3
Running time	2m:28s
Core file checked	59418
Core file modified	36
% of files modified	0.06%
Checked modules	59
PHP errors found	235
PHP warnings found	112
PHP critical errors found	83
GraphQL critical errors found	0
GraphQL warnings found	0
Total errors found	235
Total warnings found	112
Total critical errors found	119
Complexity score	975

*Example summary report

4. Tips & Advice

All issues identified by the tool are listed on the report with specific error codes. Use the [error message reference](#) docs to get more detail about each issue. Adobe also gives suggestions to fix each issue type so you can plan out your remediation steps.

Use the report to estimate the amount of effort it will take to update your code for the upgrade. Based on your experience, you will be able to estimate the required effort to upgrade based on the total number of issues identified and the severity of the issues. Since this is a command line tool, you can incorporate this into automated testing and code check suites and use the JSON output to generate your reports.

Saving the results from each upgrade project is recommended so that you can compare future upgrade results with previous results. With continued use, you'll start to develop a good sense of the level of effort it'd take to upgrade to the next version just from the summary report provided by the Upgrade Compatibility Tool.

It's also recommended that you run the tool regularly while working on the upgrade to have visibility into your progress. The number of issues should decrease as you fix them. This will also help your team decide on the best approach to distribute work.

Future releases of the tool will incorporate PHP 8.1 compatibility tests and autofixes to help you fix issues as fast as possible.

4. Implementing the upgrade

It's important to test all steps on a non-production instance before performing this in production. Follow the below steps to implement the upgrade. If needed, additional support can be found on DevDocs.

1. Switch to maintenance mode

```
bin/magento maintenance:enable
```

2. Back up composer.json file

```
cp composer.json composer.json.bak
```

3. Install composer update plugin & update dependencies

```
composer require magento/composer-root-update-plugin ~2.0  
--no-update
```

4. (Optional) Remove previous package if updating from community to commerce

```
composer remove magento/product-community-edition  
--no-update  
composer require magento/product-enterprise-edition 2.4.3  
--no-update
```

5. Update Metadata

Update the name, version, and description fields in the composer.json file as needed.

6. Update and apply dependencies & changes

```
composer update
```

7. Update database schema and data

```
bin/magento setup:upgrade
```

8. Upgrade modules and themes

Perform these for each of your third-party modules.

```
composer update vendor/module-name
```

Once complete, update the database and schema for your new code.

```
bin/magento setup:upgrade
```

9. Deploy static content and clean the cache

```
bin/magento setup:static-content:deploy  
bin/magento cache:clean
```

10. (Optional) Restart Varnish

```
service varnish restart
```

11. Disable maintenance mode

```
bin/magento maintenance:disable
```



Upgrade Best Practices

- Test all upgrade actions in a non-production environment ahead of time, to ensure there are no unforeseen issues.
- Perform your updates on a new server instance, then cut over to the new server when going live. This helps reduce downtime and catch issues before they occur.
- Leverage the [Upgrade Compatibility Tool](#) to assess the issues and complexity of your upgrade. It can also be used to actively develop new features and get ahead of issues before they arise by using [Magento PHPStorm plugin](#).
- **Automate as much testing as possible.** Testing rivals active development in the amount of time it takes as a part of the upgrade process. Leveraging automated tools provided by Adobe and third-parties will save many hours of expensive manual testing during each upgrade. Check out the [Application Testing Guide](#) for more info.
- **Develop with upgrades in mind.** Make sure to always follow best practices. Keep custom code out of the core code. Use APIs as much as possible. Document your code as you write. Use as much native functionality as you can before you customize.
- **Third-party modules** are not just assets – they are relationships. Limit the number of extensions and use only reputable companies from the [Adobe Commerce Marketplace](#). Look for a provider that can provide most/all the extensions you need. This will reduce module conflicts and ensure they work together.
- Adobe recommends an annual inventory and site cleanup of code and extensions by asking questions such as:
 - Do you still need everything?
 - Have some extensions / customizations become obsolete due to Adobe Commerce functionality expansion or shifting business priorities?
 - Is it essential to the customer experience?
 - Have you had quality or performance issues with any of your extensions? If so, should you consider removing them or finding a replacement?
 - If you are considering customizations / extensions for the upcoming year, are you considering lifetime costs, not just implementation?
- You can use this table to help you organize your extensions and customizations

#	Extension / Module Name	Composer package	Vendor	Current version	Functionality	Status	Compatible with latest AC version?	Issues	Native in AC?	Action	Notes
1	Extension name and link	extension/extension-magento-2	Vendor name	Version installed	Business requirements	Enabled / Disabled	Yes/No	List identified issues faced with this extension	Yes/No	Keep/Replace/Remove	



- **Plan and allocate resources for upgrades.** Have a 3-year rolling plan that you discuss annually at minimum and ensure the next year is planned and budgeted for. Use [Adobe's release calendar](#) for keep track of release dates. Review the [Upgrade plan checklist](#) to plan for your upgrade. The checklist is also provided at the end of this guide.
- **Know your team.** Understand who works on your projects. Is it different people each upgrade? Are they Adobe Commerce-certified? You want those with deep product and project knowledge to be on the team who manages upgrades.

Upgrade Resources

Release Strategy	Planning	Upgrade Compatibility Tool	Security	Technical
<ul style="list-style-type: none">• Release Strategy Blog• Lifecycle policy• Release schedule• Release Strategy FAQ• Release Types Devdocs• Partner Enablement Recording• New Marketplace Compatibility Policy	<ul style="list-style-type: none">• 2.4.4 Beta Program Devblog• 2.4.4 Platform Changes• Upgrade Plan Checklist• Upgrade Best Practices Guide	<ul style="list-style-type: none">• UCT Devblog• UCT Devdocs• Install UCT• PHPStorm & UCT	<ul style="list-style-type: none">• Security Scan Tool• About security patch releases blog	<ul style="list-style-type: none">• 2.4 Release Notes• Upgrade Overview Devdocs• Software Availability• Backward Compatible Development• Accessing Pre-Releases• Versioning Schema• Quality Patch Tool• Hotfixes Knowledge Base• Adobe Commerce Marketplace• Post-Launch Steps



Adobe Commerce Recommended 2022 Upgrade Paths

An eCommerce implementation is an evolution – it's never truly finished. Your business must remain one step ahead of trends by introducing the latest capabilities and functionality that will keep your customers engaged. Over time, these additional capabilities increase the footprint and overall complexity of the implementation.

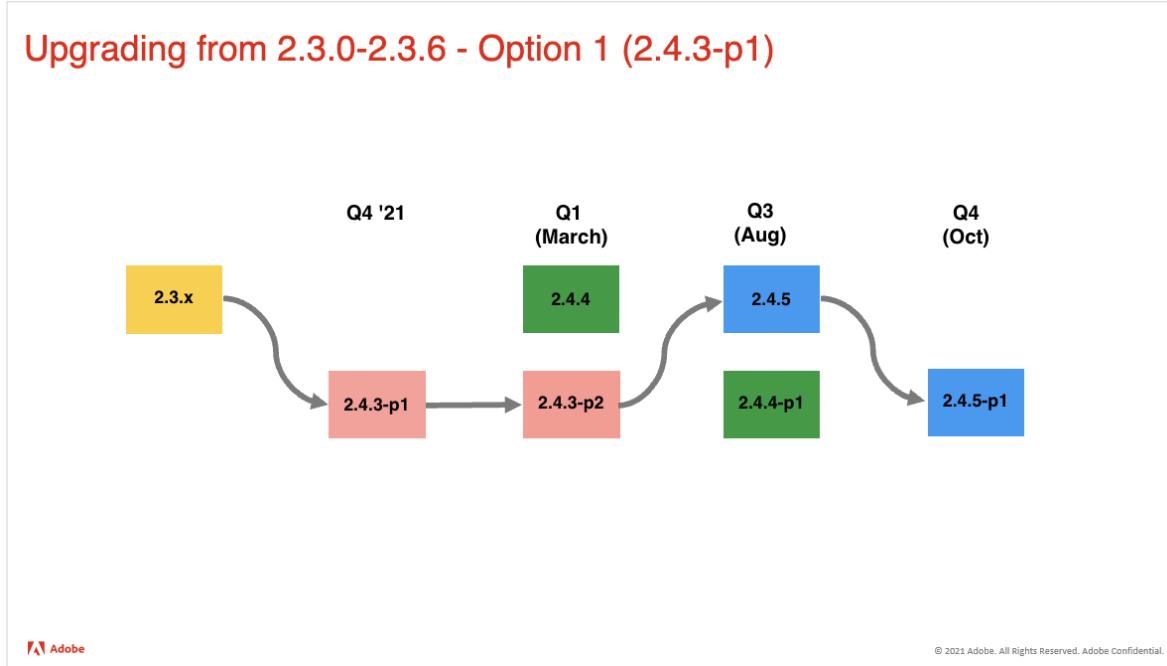
Some of the general factors that will affect the level of effort needed for your upgrade project include, but are not limited to:

Technical Complexity	Upgrade Planning and Strategy
<ul style="list-style-type: none">• Extent of customizations• Number of extensions• Number of integrations to third-parties (OMS, ERP, etc.)• Coding to best practices	<ul style="list-style-type: none">• Clarity of requirements, wavering decisions, & scope creep• Your upgrade frequency• Your testing strategy

The following are Adobe Commerce recommended paths to keep your site secure and performant throughout 2022.

Upgrading from versions 2.3.0-2.3.6

Option 1



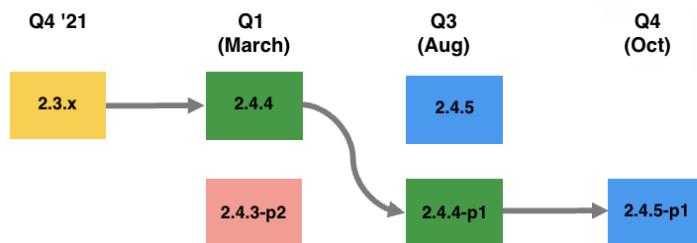
You can go from any 2.3.x line to 2.4.3. However, the longer you've gone without an upgrade, the more effort it will be to go directly to 2.4.3 as the code base will have changed more.

For example, if you are on 2.3.4 that released in January 2020, you are on a release that is almost 2 years old so the code base of 2.4.3 when compared to 2.3.4 will be far greater. This is why Adobe recommends that you upgrade often as the level of effort tends to be even higher if you delay your upgrade for a prolonged period of time.

Once you are on 2.4.3, then in Q1 '22 you can continue to be secure by upgrading to 2.4.3-p2, which will be a low effort since it's a light security release. Then in Q3 '22, you can take the full patch 2.4.5 and one more light security patch in Q4 '22 to stay secure. This path will require two high effort upgrades by the end of 2022.

Option 2

Upgrading from 2.3.0-2.3.6 - Option 2 (2.4.4)



© 2021 Adobe. All Rights Reserved. Adobe Confidential.

Alternatively, you can also upgrade from 2.3.x directly to 2.4.4 in Q1 '22. From 2.4.4, you can then take the light security patch in Q3 '22, then upgrade to the 2.4.5-p1 version in Q4 '22, which includes all updates that were in the 2.4.5 release as well as additional security patches.

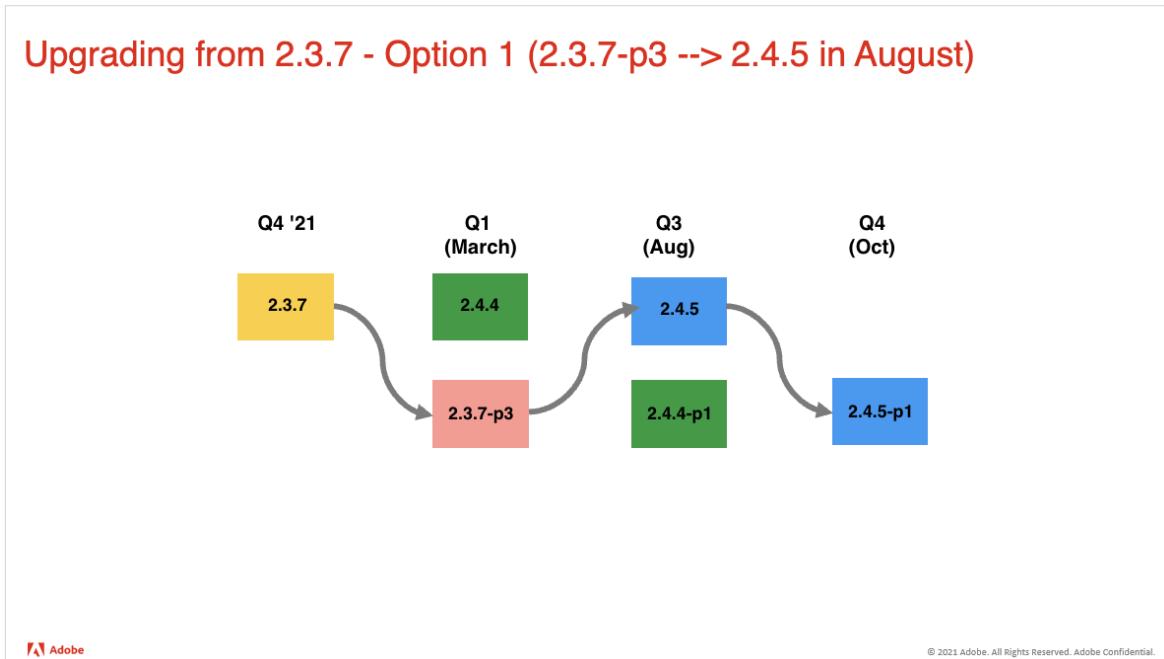
Key considerations when deciding between these two options:

Option 1: Upgrade to 2.4.3-p1 or -p2	Option 2: Upgrade to 2.4.4 or 2.4.4-p1
<ul style="list-style-type: none">Requires 2 significant upgrades before the end of 2022 to remain secure, PCI compliant, and receive Quality SupportEnables you to potentially get to a supported, PCI-compliant version soonerTiming consideration: can delay moving to a new PHP version until later in the year (August)	<ul style="list-style-type: none">Requires 1 significant upgrade and 1 low-medium effort upgrade before the end of 2022 to remain secure, PCI-compliant, and receive Quality SupportPotentially face a longer window until you get to a supported, PCI-compliant version since the 2.3 line reaches EOL in April '22Timing consideration: can start moving to a new PHP version earlier in the year (March)



Upgrading from 2.3.7

Option 1



As you are on 2.3.7, you are on a line that is only getting security releases. In Q1 '22, Adobe will release the last release of the 2.3 line, which is 2.3.7-p3, along with a security release, 2.4.3-p2 and a full release 2.4.4.

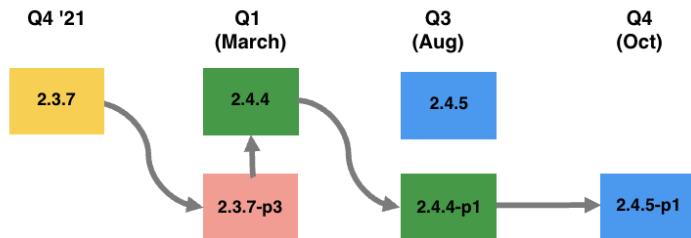
Your first option would be to take the 2.3.7-p3 and get the latest security fixes. Then, in Q3 '22, you could take the 2.4.5 release. Lastly, in Q4 '22 you could take the light security patch release based on the full 2.4.5 release. In this scenario, you would be on an EOL version for a few months until you take 2.4.5.

2.3.7 would EOL in April '22, so you would not have quality support between then until you upgrade to 2.4.5, but you would have the most recent security vulnerabilities patches from taking 2.3.7-p3.



Option 2

Upgrading from 2.3.7 - Option 2 (2.3.7-p3 --> 2.4.4)



© 2021 Adobe. All Rights Reserved. Adobe Confidential.

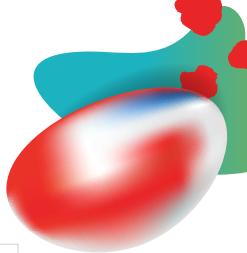
Your second option would be taking the 2.3.7-p3 release to get the latest security fixes quickly in Q1 '22, since security patches for your current line are a low effort to implement, then you can start the upgrade to 2.4.4.

In Q3 '22, you could then take 2.4.4-p1, which would be a light security release, then in Q4 '22 take 2.4.5-p1, which will include all updates included in 2.4.5 as well as the latest security releases.

You also have the option to go from 2.3.7-p3 to 2.4.4-p1, but note that the 2.4.4-p1 will be a heavy lift since you will essentially be getting all the updates included in 2.4.4 as well as the security updates in 2.4.4-p1. Deciding whether you want to start this heavier lift to the 2.4.4 line in Q1 or Q3 will be up to you and your team.

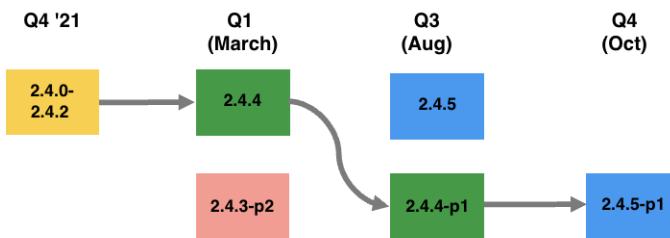
Key considerations when deciding between these two options:

Option 1: Upgrade to 2.3.7-p3 then 2.4.5	Option 2: Upgrade to 2.3.7-p3 then 2.4.4
<ul style="list-style-type: none">Receive the latest security updates with a low effort security patch firstRequires 1 significant upgrade before the end of 2022 to remain secure, PCI-compliant, and receive Quality SupportFace a longer window until you get to a supported, PCI-compliant version as 2.3 line reaches EOL in AprilTiming considerations: can delay higher effort upgrade until August	<ul style="list-style-type: none">Receive the latest security updates with a low effort security patch firstRequires 1 significant upgrade and 1 low-medium effort upgrade before the end of 2022 to remain secure, PCI-compliant, and receive Quality SupportTiming considerations: can start higher effort upgrade in March then take on a low-medium upgrade in October



Upgrading from 2.4.0-2.4.2

Upgrading from 2.4.0-2.4.2

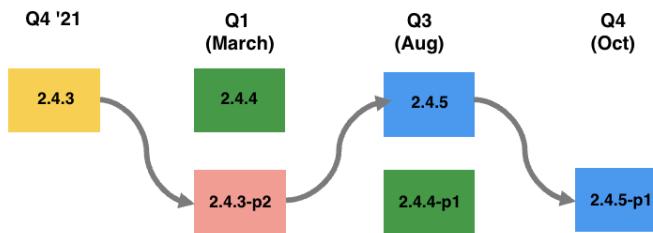


© 2021 Adobe. All Rights Reserved. Adobe Confidential.

Since you are on 2.4.0-2.4.2, we recommend that you upgrade to 2.4.4 in Q1 '22. This will be a relatively high effort due to the breaking changes in 2.4.4 caused by the move to PHP 8.1. However, the remainder of the upgrades for the year will be lower efforts, so you will only need to conduct one higher level effort upgrade in 2022.

Upgrading from 2.4.3

Upgrading from 2.4.3



© 2021 Adobe. All Rights Reserved. Adobe Confidential.

Since you are on 2.4.3, taking the security patch 2.4.3-p2 in Q1 '22 would be a low lift. Then in Q3 '22, you could take the 2.4.5 release. Lastly, in Q4 '22 you could take the light security release based on the full 2.4.5 release. In this scenario, you would only be conducting one higher level effort upgrade (upgrading to 2.4.5) in 2022.



Upgrade Plan Checklist

Goals

Review current KPIs and adjust as needed.

Extensions & Customizations

Review all current extensions and customizations and ensure they are still needed based on business requirements.

Consider replacing any extensions that don't have a good track record of keeping up-to-date with Adobe Commerce versions.

Team

Ensure, you have the right team in place, with the proper Adobe Commerce certifications and skillset.

Budget & Timing

Use the Adobe Commerce [release calendar](#) to plan your next upgrade and prepare ahead of time.

Discuss which versions you'll adopt next year (full and security patches) based on anticipated needs.

Set aside budget and team capacity for the upgrade.

3rd Party Integrations

Review current Adobe Commerce third-party integrations you have and their maintenance windows for the year. You may want to align your upgrades together.

Scope & Deployment Planning

Early access activities

- › Partner participates in [Beta program](#)
- › Review Beta release notes

Agree on budget, timeline, scope.

Run the [Upgrade Compatibility Tool](#) to identify potential issues prior to upgrading.

Consider using the upgrade to address issues identified by the [Site Wide Analysis Tool](#).

Document dependencies and any technical stack changes required such as PHP or Elastic Search versions.

Gather project team with Adobe Commerce certifications.

Create a stakeholders communications plan.

Plan maintenance window if downtime is anticipated.

Review and approve the testing strategy; use the [Application Testing Guide](#) or a third-party automation suite.

Confirm all extensions / customizations are compatible.

Review [post-launch steps](#) if issues are discovered during or after upgrade.

Post Deployment

Monitor site for issues – performance, order processing, analytics, and others.

Perform Adobe Commerce [security scan](#) or other third-party scan and review potential security vulnerabilities.

Perform a retrospective with all stakeholders and document what went well and what didn't and how to improve.

Modify your plan for the next upgrade with lessons learned.



© 2021 Adobe. All rights reserved.

Adobe, the Adobe logo, and Magento Commerce are either registered trademarks or trademarks of Adobe in the United States and/or other countries.