magmodules

How to Update a Magento 2 Extension

for

Keeping your extensions up to date ensures you have the latest features, bug fixes, and security patches. This guide covers updating extensions installed via Composer - the standard method for Magento 2.

Before You Update

1. **Backup your database** - Always have a recent backup before updates
2. **Check the changelog** - Review what's changed in the new version
3. **Test in staging first** - Never update production without testing

Update a Specific Extension

To update a single extension to the latest version:

```
composer update vendor/package-name
bin/magento setup:upgrade
bin/magento setup:di:compile
bin/magento setup:static-content:deploy
bin/magento cache:flush
```

Example - Updating Google Shopping:

```
composer update magmodules/magento2-googleshopping
bin/magento setup:upgrade
bin/magento setup:di:compile
bin/magento setup:static-content:deploy
bin/magento cache:flush
```

Update to a Specific Version

If you need a specific version rather than the latest:

```
composer require vendor/package-name:1.2.3
bin/magento setup:upgrade
bin/magento setup:di:compile
bin/magento setup:static-content:deploy
bin/magento cache:flush
```

Update All Extensions

To update all Composer packages at once:

```
composer update
bin/magento setup:upgrade
bin/magento setup:di:compile
bin/magento setup:static-content:deploy
bin/magento cache:flush
```

Tip: This also updates Magento core if a new version is available. For more control, update packages individually.

Finding Your Package Name

Not sure of the package name? Check your `composer.json` or run:

```
composer show | grep magmodules
```

Package names are different from module names:

- **Module name:** `Magmodules_GoogleShopping` (used with bin/magento commands)
- **Package name:** `magmodules/magento2-googleshopping` (used with Composer)

Check Available Updates

See which packages have updates available:

```
composer outdated
```

For Magmodules packages specifically:

```
composer outdated | grep magmodules
```

Troubleshooting

Version constraint conflicts

If Composer reports conflicts, check your version constraints:

```
composer why-not vendor/package-name 2.0.0
```

This shows which packages are blocking the update.

Memory errors

Composer can be memory-intensive. Increase PHP memory:

```
php -d memory_limit=-1 /usr/local/bin/composer update vendor/package-name
```

Class not found errors after update

Regenerate the autoloader and compiled classes:

```
composer dump-autoload
bin/magento setup:di:compile
```

Static content errors

Force static content deployment with locale:

```
bin/magento setup:static-content:deploy en_US nl_NL -f
```

Production Mode Deployment

For production environments, use maintenance mode during updates:

```
bin/magento maintenance:enable
composer update vendor/package-name
bin/magento setup:upgrade
bin/magento setup:di:compile
bin/magento setup:static-content:deploy
bin/magento cache:flush
bin/magento maintenance:disable
```

Need More Help?

Documentation:

- All Help Articles - Complete documentation overview

Support:

- Contact Support - Get help from our team