



CSV Import/Export

Customer Documents Documentation

Guide for Magento administrators, technical project owners, and implementation teams.

LAST UPDATED
2026-09-26

SOURCE
[Open this article online](#)

EXTENSION
[Magento 2 Extension Customer Documents](#)

CSV Import/Export

The CSV Import/Export add-on integrates Customer Documents with Magento's standard **Import** and **Export** screens. It also provides a command-line importer for repeatable data transfers.

- Composer package: `mageb2b/customerdocuments-importexport`
- Magento module: `MageB2B_CustomerDocumentsImportExport`
- Import and export entity: **Customer Documents**
- CLI command: `customerdocuments:import-customerdocuments`

Installation

Install the add-on in addition to the base Customer Documents extension:

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
composer require mageb2b/customerdocuments-importexport:*
php bin/magento module:enable
MageB2B_CustomerDocumentsImportExport
php bin/magento setup:upgrade
php bin/magento cache:flush
```

Production-mode stores normally also require dependency injection compilation and static-content deployment. Follow the store's normal release process.

Verify the installation

1. Open **System > Data Transfer > Export** and confirm that **Customer Documents** is available as an entity.
2. Open **System > Data Transfer > Import** and confirm the same entity is available.
3. Download the sample CSV from the Import screen.
4. Validate a small representative file before planning a bulk transfer.

Export documents

The export contains document metadata and its portable assignment values. It does not bundle the actual document files.

1. Go to **System > Data Transfer > Export**.
2. Select **Customer Documents** as the entity type.
3. Apply any required filters.
4. Click **Continue** and download the generated CSV.

The file includes document properties, customer emails, customer-group codes, category names, tags and product assignments. Product assignments use `sku@store_code@position` entries separated by `|`. Use stored paths to locate local files under `pub/media/`.

Import documents

Prepare the source file

Start with an export or the sample file from the installed module version. For a new local document, `path` must reference a real file relative to the Magento root, for example:

```
var/import/customerdocuments/northstar-safety-certificate.pdf
```

The importer copies that file into the configured Customer Documents upload folder and stores the resulting document path. When `id` identifies an existing document, `path` is optional, so metadata and assignments can be updated without uploading the file again.

Import through Magento Admin

1. Go to **System > Data Transfer > Import**.
2. Select **Customer Documents** as the entity type.
3. Choose the import behavior and upload the CSV.
4. Click **Check Data** and resolve every reported validation error.
5. Run the import and review the resulting documents and assignments.

Import through the command line

```
php bin/magento customerdocuments:import-customerdocuments \  
var/import/customer-documents.csv \  
--behavior=append \  
--delete_file_after_import=0
```

Run the command as the operating-system user used for other Magento commands. Keeping `--delete_file_after_import=0` is safer while a feed is being developed because the command deletes a successfully imported source file by default.

Import behaviors

Behavior	Purpose
<code>append</code> or <code>add_update</code>	Create records without an <code>id</code> and update existing records identified by <code>id</code>
<code>replace</code>	Delete records identified by <code>id</code> and import their replacement rows

Behavior	Purpose
<code>delete</code>	Delete records identified by <code>id</code>

Test `replace` and `delete` only in a recoverable non-production environment before using them with business documents.

Validation rules

The importer checks:

- required values such as `name`, `description` and the `path` of a new local document;
- whether customer emails and customer-group codes exist;
- whether the named category exists;
- product assignment syntax;
- whether a referenced local file exists;
- whether its extension is allowed by `customerdocuments/general/allowed_extensions`.

CSV format

Use UTF-8 with LF or CRLF line endings. The default delimiter is a comma and values may use double-quote enclosure.

```
"name","description","password","path","is_public","max_downloads",  
"customer_group_ids","customer_ids","created_at","category_id",  
",","show_in_sidebar","valid_from","valid_to","send_email","tags",  
"product_links"  
"Northstar Safety Certificate","Certificate for approved  
equipment","", "var/import/customerdocuments/northstar-safety-  
certificate.pdf","0","0","Wholesale","purchasing@northstar-  
industrial.example","", "Certificates","0","2026-01-01","2026-12-  
31","1","safety,certificate","NW-CHAIR-410@default@10"
```

Important columns:

Column	Meaning
<code>id</code>	Optional existing document ID used for updates, replacement or deletion
<code>name</code> , <code>description</code>	Required document metadata
<code>path</code>	Required for a new local document; optional when updating existing metadata
<code>customer_ids</code>	Comma-separated customer email addresses

Column	Meaning
<code>customer_group_ids</code>	Comma-separated customer-group codes
<code>category_id</code>	Category name. Despite the column name, import and export use the readable category name rather than its numeric database ID.
<code>send_email</code>	<code>1</code> queues assignment notifications after import
<code>tags</code>	Comma-separated tag names
<code>product_links</code>	<code>sku@store_code@position</code> entries separated by <code> </code>

CLI options include `--behavior`, `--field_separator`, `--field_multiple_value_separator`, `--fields_enclosure` and `--delete_file_after_import`.

Automation

The package does not schedule imports itself. Use the store's deployment or operations scheduler when an upstream system produces a file:

```
php /path/to/magento/bin/magento customerdocuments:import-  
customerdocuments \  
  /path/to/customer-documents.csv \  
  --behavior=append \  
  --delete_file_after_import=0
```

Before running unattended imports:

1. Validate a representative file manually.
2. Decide where successful and failed files are retained.
3. Monitor the command's exit code and output.
4. Verify imported access assignments with a fictional customer in the correct website.
5. Keep a recoverable copy outside Magento's import directory.

Related guides

- [Customer Documents configuration](#)
- [Access control](#)
- [Common issues](#)