Installation

Punchout Documentation

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

LAST UPDATED
2026-10-13

SOURCE
Open this article online

EXTENSION
Magento 2 Extension Punchout

Installation

Install the module first, then create one buyer profile and one outbound mapping profile. Do not connect a production procurement account until the login, cart return and retry behavior have been tested end to end.

Requirements

- Magento 2.4
- PHP `8.1` or newer within the PHP version supported by your Magento release
- HTTPS for Magento and every production return URL
- Magento cron running for cleanup and document-queue processing
- Magento Inventory (MSI) modules enabled; `MageB2B_Punchout` requires `Magento_InventoryApi` and `Magento_InventorySalesApi`, which supply the salable quantity for OCI `QUANTITYCHECK` responses

Install the package

Run these commands from the Magento project root:

```
composer require mageb2b/punchout
php bin/magento module:enable MageB2B_ExtensionManager
MageB2B_Punchout
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento cache:clean
```

Deploy static content when your release process requires it:

```
php bin/magento setup:static-content:deploy -f
```

Confirm the admin areas

Open **Punchout** in the main Magento Admin navigation. The menu contains:

- Punchout Buyers
- Punchout Mappings
- Punchout Trace
- Document Queue
- Inbound Orders
- Analytics
- Punchout Simulator
- Configuration

If an entry is missing, check the administrator role under **System > Permissions > User Roles**. PunchOut has separate permissions for buyers, mappings, traces, inbound orders, the document queue, analytics, the simulator and configuration.

Create the first working profile

1. Open **Punchout > Punchout Mappings**.
2. Add an outbound profile for the required protocol.
3. Select a matching starter template, save the copied rules, and keep the profile active.
4. Open **Punchout Buyers** and add the external buyer identity.
5. Enter the OCI credentials or cXML sender credentials supplied by the procurement team.
6. Select the customer-resolution mode and assign the new outbound mapping profile.
7. Keep the buyer inactive until the values have been reviewed.

Use a buyer identifier that belongs to the integration, such as `NORTHWIND-PROCUREMENT-EU`, rather than a person's name or an environment password.

Run the first test

Enable trace logging and use **Punchout Simulator** to build the request. A useful first test proves all of the following:

- the buyer is resolved in the intended store view;
- protocol and optional transport credentials are accepted;
- a context token and correlation ID are created;
- the storefront session opens with the expected guest or customer context;
- the cart is returned with the fields required by the buyer;
- the trace contains sanitized request and response information.

For cXML projects that create Magento orders, continue with one approved `OrderRequest`. Confirm the effective inbound processing mode, returned-cart price policy, customer and product resolution, shipping, payment and exact-retry behavior before go-live.

Uninstall the module

Uninstalling `MageB2B_Punchout` runs a destructive cleanup: every `punchout_*` table is dropped (buyer profiles, mappings, session contexts, traces, returned-cart snapshots, inbound orders, document-queue rows and analytics data) and all `core_config_data` entries under `punchout/` and `payment/punchout_purchaseorder/` are removed. The operation is not reversible, so export any data you still need before running `bin/magento module:uninstall`.

The `mageb2b/punchout-iframe` add-on ships its own uninstall step that drops the

`punchout_iframe_buyer_config` table and removes the `punchout/iframe/` configuration values.

Continue with Configuration before exposing the endpoints to a production buyer.