JAGGAER cXML Setup

Punchout Documentation

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

LAST UPDATED
2026-11-05

SOURCE
Open this article online

EXTENSION
Magento 2 Extension Punchout

JAGGAER cXML Setup

Use `jaggaer_cxml_outbound_core` as an outbound mapping starter for a JAGGAER buyer. It copies a small set of cart fields into an editable runtime profile. It is not JAGGAER certification or a replacement for the customer's integration specification and acceptance tests.

The template must be present in the installed PunchOut package. Magento synchronizes the reusable mapping templates during the normal `bin/magento setup:upgrade` deployment step. Updating a reusable template does not rewrite existing customer runtime mappings.

Confirm the buyer's contract

Ask the procurement team for its JAGGAER product, current integration specification, test account, required fields and document endpoints. Confirm the cXML version separately for cart return, purchase orders, confirmations, shipments and invoices. The starter's metadata references the documented 1.1.010 PunchOut baseline; it does not force every transaction to that version.

JAGGAER publishes a Supplier Integration Specification through its Supplier Reference Materials. The publicly linked specification is dated July 2023, product version 23.2. Confirm whether the customer's current contract differs.

Create the runtime mapping

1. Open **Punchout > Punchout Mappings** and add a profile.
2. Select the buyer's Store View, protocol **cXML**, direction **Outbound**, and a unique profile code.
3. Leave **Reusable Template Profile** disabled.
4. In **Start From Template**, select **JAGGAER cXML Outbound Core**.
5. Review the copied rules, adapt customer-required fields, and save the active profile.

Returned field	Source and behavior
Quantity	Actual cart quantity, formatted to three decimal places
Supplier part ID	Cart SKU; required
Unit price	Actual cart price, formatted to two decimal places; an explicit zero remains zero
Currency	Resolved PunchOut context currency; no EUR or USD substitution
Description	Cart description; required

Returned field	Source and behavior
Unit of measure	Cart unit with the configured unit mapping; no automatic `EA` replacement for missing data
Classification	Optional product attribute `unspsc`
Classification domain	Context classification domain, defaulting to `UNSPSC`

If the buyer requires UNSPSC, select the actual maintained product attribute, verify its values and make the classification rule mandatory. Add manufacturer name, manufacturer part number and other customer-required fields only from maintained sources. Do not substitute the SKU for a manufacturer's part number without an explicit business agreement.

The protocol adapter supplies line numbers and line-specific `SupplierPartAuxiliaryID` values. Keep those identifiers intact in the procurement return and subsequent purchase order. They distinguish configurations that share the same SKU.

Configure the buyer

Create an active cXML Buyer Profile for the intended Store View. Configure its sender domain, sender identity and shared secret, choose the customer-resolution mode, and assign the copied outbound runtime mapping. Review product visibility and negotiated prices under that exact Magento customer.

The shared setup endpoint is `/punchout/cxml/setup`. cXML sender credentials identify the buyer; the buyer-specific browser return URL comes from the authenticated setup request. Confirmation, shipment and invoice endpoints are separate settings. See Buyer and Mapping Profiles and Endpoint Reference.

BuyerCookie compatibility checkpoint

The JAGGAER specification describes a BuyerCookie that can be constant across transactions. Punchout 1.2.3 stores returned carts by their internal PunchOut session, so independent sessions can retain the same BuyerCookie. Earlier versions require an upgrade for this workflow.

Do not alter the received BuyerCookie to manufacture uniqueness. Echo the original value. For later purchase orders, preserve each line's SupplierPartAuxiliaryID exactly as supplied by Magento. An exact identifier selects the retained original cart within the authenticated buyer, store, customer and currency context, including when the PO contains no BuyerCookie. If no exact identifier matches, the missing-snapshot policy applies; neither a unique remaining BuyerCookie nor a matching SKU can select another cart. Unknown explicit line identifiers cannot inherit a different line's original price or options. Conflicting references to several retained carts are rejected; a single PO combining lines from several returned carts is not supported by the current one-cart-per-

order model.

Validate the complete workflow

Test setup, customer activation, shopping and browser cart return with the copied mapping. Inspect the actual received cXML, not only Magento's successful HTTP response. Compare quantities, units, currency, configured product options, negotiated prices and computed totals.

Also test two sessions with the same BuyerCookie, two configurations of the same SKU, Edit/Inspect, changed quantities, and a purchase order submitted after the browser session expires. Verify the saved Magento order against the approved PO and retained cart data. Configure and test the intended inbound-order price policy.

Finally create a Magento shipment and invoice for the test order and inspect their actual cXML delivery and acknowledgements. These are separate document integrations; the cart mapping alone does not enable them. See Document Loop and Queue.

Record the accepted request/response samples and customer-specific changes using sanitized test data. Keep credentials, tokens and production identifiers out of screenshots and support exports.