



SOFTWARESILOS

Punchout Documentation

PDF Manual

MODULE

punchout

LAST UPDATED

2026-03-29

Table of Contents

Punchout Documentation · punchout

- [Overview](#)
- [Punchout Documentation](#)
- [Getting Started](#)
 - [Installation](#)
 - [Configuration](#)
 - [Go-Live Checklist](#)
- [Features](#)
 - [Admin Field Reference](#)
 - [Authentication and Security Model](#)
 - [Buyer and Mapping Profiles](#)
 - [Cart API and Session Lifecycle](#)
 - [Security and Idempotency](#)
 - [Trace and Simulator](#)
- [Flows](#)
 - [cXML Flow](#)
 - [OCI Flow](#)
- [Templates Overview](#)
 - [Ariba Templates](#)
 - [Coupa Templates](#)
 - [cXML Core Templates](#)
 - [OCI and SAP Templates](#)
 - [Oracle Templates](#)
 - [Template Selection Guide](#)
- [Troubleshooting](#)
 - [Common Issues](#)
 - [Error Catalog](#)

Punchout Documentation

Punchout Documentation · /punchout

Punchout connects Magento 2 with procurement systems through OCI and cXML. This documentation covers setup, mapping, transfer flow, security, templates, and operations.

Quick Links

Getting Started

- [Installation](#)
- [Configuration](#)
- [Go-Live Checklist](#)

Flows

- [OCI Flow](#)
- [cXML Flow](#)

Templates

- [Templates Overview](#)
- [Template Selection Guide](#)
- [OCI and SAP Templates](#)
- [cXML Core Templates](#)
- [Ariba Templates](#)
- [Coupa Templates](#)
- [Oracle Templates](#)

Features

- [Admin Field Reference](#)
- [Authentication and Security Model](#)
- [Buyer and Mapping Profiles](#)
- [Cart API and Session Lifecycle](#)
- [Security and Idempotency](#)
- [Trace and Simulator](#)

Troubleshooting

- [Common Issues](#)
- [Error Catalog](#)

Installation

Punchout Documentation · /punchout/getting-started/installation

This page gives a practical installation baseline for a first production-ready Punchout setup.

1. Install and enable the module

Install `MagentoB2B_Punchout` in your Magento instance and run your standard Magento setup commands:

1. `module enable`
2. `setup upgrade`
3. `cache clean`

If your environment requires compilation, run DI compile before final testing.

2. Verify required routes

Confirm that these routes are available:

- `/punchout/oci/login`
- `/punchout/cxml/setup`
- `/punchout/oci/transfer`
- `/punchout/cxml/transfer`
- `/punchout/health/ping`

3. Verify admin menu entries

Confirm these admin sections are visible and accessible:

- Punchout Buyers
- Punchout Mappings
- Punchout Trace
- Punchout Simulator
- Stores > Configuration > Punchout

If a menu is missing, verify ACL role permissions for the current admin user.

4. Set base configuration

In admin configuration, set:

- `Punchout Enabled = Yes`
- `Sandbox Mode = Yes` for non-production, `No` for production
- security options (signature mode, signature secret, IP policy)
- trace retention policy (recommended 14 to 30 days for rollout)

5. Create the first buyer and mapping profile

Create one buyer profile and one mapping profile for your first target protocol. Link the mapping profile

to the buyer.

Recommended first setup:

- protocol: `oci` or `cxml`
- buyer id: external procurement identity, for example `BUYER_TEST_001`
- mapping profile: start from template, then adjust only partner-specific fields

6. Validate with simulator

Run one simulator scenario and verify request payload, signature behavior, and expected transfer output.

Minimum acceptance checks:

- inbound request is authenticated
- punchout context token is created
- cart transfer returns expected mapped fields
- trace entry contains correlation id and endpoint details

7. Continue to go-live hardening

After the first successful simulation, continue with:

- [Configuration](#)
- [Go-Live Checklist](#)

Configuration

Punchout Documentation · /punchout/getting-started/configuration

Punchout configuration is split into global settings, buyer profile settings, mapping profile settings, and runtime security controls.

1. Global settings (Stores > Configuration > Punchout)

Recommended baseline:

- Punchout Enabled = Yes
- Sandbox Mode = Yes in test environments only
- Strict Signature Check = Yes in production
- Allowed IP Ranges limited to your procurement platform egress IPs
- Trace Enabled = Yes during integration and go-live
- Trace Retention Days = 14 to 30

2. Buyer profile configuration

A buyer profile defines who can start Punchout and under which protocol identity.

Required fields:

- Active = Yes
- Protocol = oci or cxml
- Buyer ID = external procurement identity
- Store View = specific scope or All Store Views
- Mapping Profile = selected profile for transfer behavior

Credential handling:

- use typed credential fields in admin
- secrets are stored encrypted internally
- avoid sharing one buyer profile across unrelated customers

3. Mapping profile configuration

A mapping profile controls how source data becomes outbound transfer fields.

Required structure:

- protocol and direction must match the buyer and flow
- source entity and source field
- target field
- required flag
- sort order

Recommended structure:

- start from a template profile
- keep partner-specific overrides in a dedicated custom profile
- avoid editing template profiles directly for production projects

4. Runtime security behavior

Security checks are evaluated in sequence during inbound requests.

Common settings:

- strict signature mode
- signature secret
- IP allowlist
- hook URL validation behavior

5. Operational controls

Recommended runtime controls:

- trace logging enabled in integration and go-live phases
- retention tuned to support/debugging needs
- sandbox mode only for non-production endpoints and tests

6. Configuration order for first rollout

1. Configure global security and trace options.
2. Create buyer profile with protocol and identity.
3. Create mapping profile from template and assign it.
4. Execute simulator and verify trace details.
5. Enforce strict mode and narrow IP ranges before production.

Go-Live Checklist

Punchout Documentation · </punchout/getting-started/go-live-checklist>

Use this checklist before moving Punchout from test to production.

Security

- Punchout Enabled = Yes
- Sandbox Mode = No
- strict signature validation enabled
- signature secret rotated and documented
- IP allowlist restricted to trusted source ranges
- HTTPS endpoints verified for inbound and outbound routes

Buyer and Mapping

- every active buyer has exactly one intended protocol
- buyer identities verified against procurement platform data
- mapping profile assigned and tested with real sample payloads
- required mapping fields populated for the selected template variant
- no temporary test buyer remains active

Runtime Reliability

- idempotency behavior tested for retry and conflict scenarios
- trace logging enabled for go-live window
- trace retention set to support issue triage
- alerting owner defined for failed transfer requests

Functional Validation

- OCI login flow success
- cXML setup flow success (if used)
- cart transfer payload validated by procurement side
- purchase order document loop validated (if enabled in project scope)

Operational Readiness

- support team has trace export procedure
- integration partner contact list documented
- rollback plan prepared (disable specific buyer or protocol quickly)

Recommended Sign-Off

1. Integration owner sign-off
2. Magento admin owner sign-off
3. Procurement platform owner sign-off

Admin Field Reference

Punchout Documentation · </punchout/features/admin-field-reference>

This page describes the main Punchout admin fields in business-friendly language.

Buyer Profile Fields

Store View

- Purpose: scope buyer behavior to a specific store view.
- Recommended: use store-specific scope when price catalogs differ by store.

Buyer ID

- Purpose: identity key sent by the procurement platform.
- Recommended: use a stable external identifier, for example `COMPANY_A_BUYER_01`.

Protocol

- Options: `oci`, `cxml`
- Purpose: defines the expected inbound/outbound request model.

Active

- Purpose: enable or disable this buyer quickly without deleting configuration.
- Recommended: set to `No` for deprecated or test buyers.

Mapping Profile

- Purpose: links transfer field rules to this buyer.
- Recommended: assign one profile per integration variant and keep names explicit.

Mapping Profile Fields

Name and Code

- Purpose: unique identification in admin and logs.
- Recommended format: include target system and direction, for example `coupa_outbound_main`.

Protocol and Direction

- Purpose: determine compatible flows and templates.
- Rule: must match buyer protocol and intended endpoint behavior.

Start From Template

- Purpose: preload proven mapping rules.
- Recommended: use templates for first rollout, then apply minimal custom changes.

Mapping Rules

- Source Entity: where data comes from (quote item, product, context).
- Source Field: Magento-side value key.
- Target Field: outbound protocol field.
- Transformer: normalization logic, such as trim or price format.
- Default Value: fallback if source is empty.
- Required: field must be present for successful transfer.
- Sort Order: evaluation order for deterministic output.

Global Configuration Fields

Punchout Enabled

- Purpose: master switch for module behavior.

Sandbox Mode

- Purpose: separates test and production behavior.
- Recommended: disable in production.

Strict Signature Check

- Purpose: reject unsigned or invalidly signed requests.
- Recommended: enable in production.

Signature Secret

- Purpose: shared secret for signature validation.
- Recommended: rotate periodically and store securely.

Allowed IP Ranges

- Purpose: inbound source restriction.
- Recommended: allow only procurement platform egress IPs.

Trace Enabled and Retention

- Purpose: support diagnosis and audit.
- Recommended: enable during integration, keep at least 14 days retention in rollout phase.

Authentication and Security Model

Punchout Documentation · /punchout/features/authentication-and-security-model

Punchout uses layered validation so integration errors and unauthorized requests are separated clearly.

Validation Order

1. Resolve buyer profile from inbound identity.
2. Validate protocol identity fields.
3. Validate credentials or shared secret.
4. Validate transport-level constraints (IP, signature, URL policy).
5. Create or resume Punchout context token.

Protocol Identity Check

OCI

- identity is usually bound to buyer id and configured OCI credentials.
- mismatch result: buyer not recognized or credential mismatch.

cXML

- identity is usually mapped from sender credential domain and identity fields.
- mismatch result: sender identity mismatch.

Credential Check

Credentials are configured in typed admin fields and stored encrypted internally.

Expected outcomes:

- valid credential: continue request processing
- missing credential: hard fail with actionable error
- wrong credential: authentication failure

Transport Security Check

Signature validation

- strict mode enforces signed requests.
- recommendation: strict mode enabled in production.

IP allowlist

- only allowed source networks can call inbound endpoints.
- recommendation: use exact platform egress ranges.

Hook URL validation

- return/callback URLs are validated against policy to prevent unsafe redirects.

Practical Production Baseline

- strict signature validation on
- IP allowlist on
- secrets rotated and access-controlled
- trace logging enabled with payload sanitization

Buyer and Mapping Profiles

Punchout Documentation · /punchout/features/buyer-and-mapping-profiles

Buyer and mapping profiles are the core admin entities for controlled Punchout rollout.

Buyer profiles

Buyer profiles define runtime eligibility and protocol identity.

Business value:

- separate integration behavior per buyer
- controlled activation/deactivation
- encrypted credential handling

Mapping profiles

Mapping profiles define transfer output structure through rule-based field mapping.

Business value:

- no partner-specific code branches for each mapping variant
- faster onboarding for additional procurement systems
- consistent mapping governance in admin

Template-first rollout

Profiles can be initialized from templates and then adapted. This shortens implementation cycles while keeping mapping behavior auditable.

Related pages

- [Admin Field Reference](#)
- [Template Selection Guide](#)

Cart API and Session Lifecycle

Punchout Documentation · /punchout/features/cart-api-and-session

Punchout runtime depends on context tokens and a deterministic session lifecycle.

Tokenized cart API

The module provides token-bound operations to:

- add cart items
- update quantities
- remove items
- read current cart state

This supports external procurement-driven cart orchestration.

Session context lifecycle

Each Punchout process tracks context state, including token, correlation, buyer reference, and lifecycle status.

Why this matters

A dedicated context lifecycle prevents session mix-ups between Punchout and standard storefront behavior and improves process reproducibility.

Security and Idempotency

Punchout Documentation · /punchout/features/security-and-idempotency

Punchout endpoints are protected through layered runtime validation and deterministic retry control.

Security layers

Typical checks include:

- buyer profile resolution
- buyer authentication
- IP allowlist validation
- signature validation (strict mode optional)
- hook URL validation

This reduces risk from invalid, malformed, or unauthorized requests.

Idempotency

Idempotency keys allow transfer/document endpoints to process retries safely.

Expected behavior:

- new request processed normally
- known key can return replay result
- mismatched payload can return conflict status

Business outcome:

- fewer duplicate transfers
- safer integration under network instability

Related pages

- [Authentication and Security Model](#)
- [Error Catalog](#)

Trace and Simulator

Punchout Documentation · /punchout/features/trace-and-simulator

Trace and simulator features are designed for integration support and go-live reliability.

Trace viewer

Trace entries provide:

- direction (inbound/outbound)
- endpoint and protocol context
- status and timing
- correlation id
- sanitized payload snippets

The admin UI supports detail view and JSON export.

Simulator

The simulator can generate reproducible OCI and cXML test requests in admin.

Typical outputs:

- target endpoint
- generated payload
- signature details
- cURL sample

Operational benefit

Support and integration teams can reproduce issues faster and validate fixes without ad-hoc scripting for each test run.

cXML Flow

Punchout Documentation · /punchout/flows/cxml-flow

The cXML flow supports setup and transfer with context persistence.

Step 1: PunchOutSetupRequest

The setup endpoint validates buyer, security, authentication, and hook URL constraints.

The module extracts request fields and creates a Punchout context with token and correlation.

Step 2: Optional level-2 handling

For setup requests carrying level-2 metadata, the module resolves selected-item context and can provide a deterministic deep-link hint.

Step 3: PunchOutOrderMessage transfer

The transfer endpoint builds outbound cXML payload using cart content plus mapping profile logic.

Document feedback scope

After Punchout transfer, document-level processing runs through dedicated document endpoints (PO inbound, ASN outbound, invoice outbound).

The cXML transfer route itself remains focused on cart transfer payload generation.

Operational notes

- idempotency handling is supported on transfer
- trace entries include in/out direction and correlation
- payload logging is sanitized for sensitive fields

Template references

For faster onboarding, start from vendor-specific cXML templates:

- [Templates Overview](#)
- [cXML Core Templates](#)
- [Ariba Templates](#)
- [Coupa Templates](#)
- [Oracle Templates](#)

OCI Flow

Punchout Documentation · /punchout/flows/oci-flow

The OCI flow covers login/setup, cart interaction, and outbound cart transfer.

Step 1: OCI login request

The OCI login endpoint validates buyer profile, security constraints, buyer authentication, and return URL constraints.

If successful, Punchout creates a session context with token and correlation identifier.

Step 2: Catalog and cart interaction

The user or procurement process works in Magento while the session context remains bound to the Punchout token.

Cart operations can also be driven through tokenized cart endpoints.

Step 3: OCI transfer

Transfer builds outbound payload based on current cart data and mapping rules.

The response is an OCI-compatible transfer payload (typically form-post based) that returns data to the procurement side.

Document feedback scope

Post-order document feedback is handled through dedicated document endpoints (PO inbound, ASN outbound, invoice outbound).

There is no separate generic "return-transfer" controller in the Punchout transfer path.

Reliability notes

- transfer can enforce idempotency keys
- repeated calls can return deterministic replay behavior
- failures include structured error payloads for easier handling

Template references

For production onboarding, use the template library instead of building mappings from scratch:

- [Templates Overview](#)
- [OCI and SAP Templates](#)

Templates Overview

Punchout Documentation · </punchout/templates>

Punchout ships protocol-specific template profiles so teams can start from tested defaults instead of manual mapping.

Scope

- Source fixtures are shipped in `app/code/MageB2B/Punchout/etc/mapping/templates/*.json`.
- Templates are synced into Magento on `setup:upgrade`.
- Each template has a stable code, protocol, direction, and predefined mapping rules.

Navigation

- [Template Selection Guide](#)

OCI

- [OCI and SAP Templates](#)

cXML

- [cXML Core Templates](#)
- [Ariba Templates](#)
- [Coupa Templates](#)
- [Oracle Templates](#)

Template Matrix

Template Code	Protocol	Direction	Target System	Use Case	Details
oci_outbound_core	OCI	Outbound	Generic	Standard OCI cart transfer	OCI and SAP
oci_outbound_extended	OCI	Outbound	Generic	Additional OCI extension fields	OCI and SAP
sap_oci_v4_outbound_core	OCI 4.0	Outbound	SAP	SAP OCI 4 baseline	OCI and SAP
sap_oci_v5_outbound_core	OCI 5.0	Outbound	SAP	SAP OCI 5 baseline	OCI and SAP
sap_oci_v5_outbound_hierarchy	OCI 5.0	Outbound	SAP	Parent/child and material group flow	OCI and SAP
sap_oci_v5_outbound_secure	OCI 5.0	Outbound	SAP	Fail-fast required mapping set	OCI and SAP
cxml_outbound_core	cXML	Outbound	Generic	Standard PunchOutOrderMessage	cXML Core
cxml_outbound_level2	cXML	Outbound	Generic	Optional level-2 enrichments	cXML Core
ariba_cxml_outbound_core	cXML	Outbound	SAP Ariba	Ariba baseline	Ariba

Template Code	Protocol	Direction	Target System	Use Case	Details
ariba_cxml_outbound_level2	cXML	Outbound	SAP Ariba	Ariba level-2 with extrinsics	Ariba
ariba_cxml_outbound_level2_strict	cXML	Outbound	SAP Ariba	Strict Ariba level-2 enforcement	Ariba
coupa_cxml_inbound_posr_1_2_014	cXML 1.2.014	Inbound	Coupa	Setup request normalization	Coupa
coupa_cxml_outbound_poom_1_2_023	cXML 1.2.023	Outbound	Coupa	POOM outbound baseline	Coupa
coupa_cxml_outbound_poom_1_2_023_strict	cXML 1.2.023	Outbound	Coupa	Strict POOM outbound	Coupa
oracle_cxml_outbound_1b_core	cXML 1b	Outbound	Oracle Procurement	Oracle baseline	Oracle
oracle_cxml_outbound_1b_core_strict	cXML 1b	Outbound	Oracle Procurement	Strict Oracle profile	Oracle

How to Choose

1. Pick protocol and direction first (OCI outbound, cXML inbound, or cXML outbound).
2. Pick target platform pack (generic, SAP, Ariba, Coupa, Oracle).
3. Start with core for rollout, move to strict only when source data quality is stable.

Ariba Templates

Punchout Documentation · </punchout/templates/ariba>

This page covers SAP Ariba-oriented cXML outbound templates.

ariba_cxml_outbound_core

- Direction: Outbound
- Rules: 9 (6 required)
- When to use: Ariba baseline line-item transfer.
- Key target fields: quantity, supplier part IDs, price/currency, description, UOM, and classification domain.

ariba_cxml_outbound_level2

- Direction: Outbound
- Rules: 12 (6 required)
- When to use: Ariba level-2 transfer with manufacturer and buyer extrinsic fields.
- Key target fields: Core fields plus ItemDetail/ManufacturerName, ItemDetail/ManufacturerPartID, and ItemDetail/Extrinsic[name=AribaBuyerId].

ariba_cxml_outbound_level2_strict

- Direction: Outbound
- Rules: 12 (all required)
- When to use: Controlled projects where every mapped level-2 field must be present.
- Key target fields: Same as level-2, but strict requirement flags.

Reference Sources

- SAP Ariba cXML documentation entry point:
<https://help.sap.com/docs/buying-invoicing/catalog-format-reference/cxml>
- cXML Reference Guide: <https://xml.cxml.org/current/cXMLReferenceGuide.pdf>

Coupa Templates

Punchout Documentation · </punchout/templates/coupa>

This page covers Coupa-focused cXML templates, including inbound setup normalization and outbound order transfer.

coupa_cxml_inbound_posr_1_2_014

- Direction: Inbound
- Rules: 6 (5 required)
- When to use: Normalize incoming PunchOut setup requests for Coupa-compatible cXML 1.2.014 payloads.
- Key target fields: setup operation, sender credential domain/identity, buyer cookie, browser form post URL, selected item supplier part ID.

coupa_cxml_outbound_poom_1_2_023

- Direction: Outbound
- Rules: 10 (6 required)
- When to use: Baseline outbound PunchOutOrderMessage aligned with Coupa cXML 1.2.023 usage.
- Key target fields: core item fields plus
ItemDetail/Extrinsic[name=SupplierPunchoutURL].

coupa_cxml_outbound_poom_1_2_023_strict

- Direction: Outbound
- Rules: 10 (all required)
- When to use: Coupa outbound projects that require full data completeness enforcement.
- Key target fields: same as baseline, with strict requirement flags.

Reference Sources

- Coupa supported cXML DTD versions by document type:
<https://compass.coupa.com/en-us/products/product-documentation/supplier-resources/for-suppliers/coupa-supplier-portal/set-up-the-csp/purchase-orders/purchase-order-collaboration-with-buyers/supported-cxml-dtd-version-per-document-type>
- cXML Reference Guide: <https://xml.cxml.org/current/cXMLReferenceGuide.pdf>

cXML Core Templates

Punchout Documentation · </punchout/templates/cxml-core>

This page covers protocol-generic cXML outbound templates.

cxml_outbound_core

- Direction: Outbound
- Rules: 6 (5 required)
- When to use: Standard PunchOutOrderMessage transfer for mixed cXML buyers.
- Key target fields: ItemIn@quantity, ItemID/SupplierPartID, ItemDetail/UnitPrice/Money, ItemDetail/Description, ItemDetail/UnitOfMeasure, ItemDetail/Classification.

cxml_outbound_level2

- Direction: Outbound
- Rules: 5 (all optional)
- When to use: Add optional level-2 fields on top of a core mapping profile.
- Key target fields: ItemID/SupplierPartAuxiliaryID, ItemDetail/ManufacturerName, ItemDetail/ManufacturerPartID, ItemDetail/Classification@domain, ItemDetail/Extrinsic[name=ImageURL].

Reference Sources

- cXML Reference Guide: <https://xml.cxml.org/current/cXMLReferenceGuide.pdf>
- cXML release notes: <https://xml.cxml.org/current/ReleaseNotes.html>

OCI and SAP Templates

Punchout Documentation · </punchout/templates/oci-and-sap>

This page covers all shipped OCI outbound templates.

oci_outbound_core

- Direction: Outbound
- Rules: 6 (all required)
- When to use: Generic OCI transfer without ERP-specific extras.
- Key target fields: NEW_ITEM-DESCRIPTION, NEW_ITEM-QUANTITY, NEW_ITEM-UNIT, NEW_ITEM-PRICE, NEW_ITEM-CURRENCY, NEW_ITEM-VENDORMAT.

oci_outbound_extended

- Direction: Outbound
- Rules: 5 (optional enrichment set)
- When to use: Add optional details such as material group, lead time, or long text.
- Key target fields: NEW_ITEM-ADDITIONAL_COST, NEW_ITEM-MATGROUP, NEW_ITEM-EXT_PRODUCT_ID, NEW_ITEM-LONGTEXT, NEW_ITEM-LEADTIME.

sap_oci_v4_outbound_core

- Direction: Outbound
- Rules: 7 (6 required)
- When to use: SAP environments aligned with OCI 4.0 field behavior.
- Key target fields: Core OCI fields plus NEW_ITEM-MATNR.

sap_oci_v5_outbound_core

- Direction: Outbound
- Rules: 8 (6 required)
- When to use: SAP OCI 5.0 baseline with quote correlation support.
- Key target fields: Core OCI fields plus NEW_ITEM-MATNR and NEW_ITEM-EXT_QUOTE_ID.

sap_oci_v5_outbound_hierarchy

- Direction: Outbound
- Rules: 8 (6 required)
- When to use: Parent-child or bundle-like transfer models.
- Key target fields: Includes NEW_ITEM-PARENT_ID and NEW_ITEM-MATGROUP.

sap_oci_v5_outbound_secure

- Direction: Outbound
- Rules: 8 (all required)
- When to use: Fail-fast profile for controlled production environments.
- Key target fields: Includes NEW_ITEM-EXT_QUOTE_ID and NEW_ITEM-EXT_CONTRACT.

Reference Sources

- SAP OCI 4.0 documentation:
https://help.sap.com/doc/srm/1.0/en-US/attachments/3362936504/OCI_4.0_Documentation.pdf
- SAP OCI 5.0 documentation:
https://help.sap.com/doc/srm/1.0/en-US/attachments/3362936504/OCI_50_EN20130121.pdf

Oracle Templates

Punchout Documentation · /punchout/templates/oracle

This page covers Oracle Procurement oriented cXML outbound templates.

oracle_cxml_outbound_1b_core

- Direction: Outbound
- Rules: 10 (6 required)
- When to use: Oracle cXML baseline with buyer identifier support.
- Key target fields: core line-item fields plus ItemDetail/Extrinsic[name=OracleBuyerId].

oracle_cxml_outbound_1b_core_strict

- Direction: Outbound
- Rules: 10 (all required)
- When to use: Oracle projects with strict transfer validation.
- Key target fields: same field set as core, but all entries are required.

Reference Sources

- Oracle cXML Punchout attributes:
<https://docs.oracle.com/en/cloud/saas/procurement/25c/oapro/attributes-supported-in-oracle-cxml-punchout.html>
- cXML Reference Guide: <https://xml.cxml.org/current/cXMLReferenceGuide.pdf>

Template Selection Guide

Punchout Documentation · </punchout/templates/selection-guide>

Use this page to choose the right template quickly.

Step 1: Choose Protocol and Direction

- OCI outbound: cart transfer in OCI field format.
- cXML outbound: PunchOutOrderMessage transfer.
- cXML inbound: setup/request normalization scenarios.

Step 2: Choose Platform Pack

Platform	Recommended Start	When to switch
Generic OCI	<code>oci_outbound_core</code>	use <code>oci_outbound_extended</code> if extra OCI fields are required
SAP OCI 4	<code>sap_oci_v4_outbound_core</code>	move to SAP OCI 5 templates when the buyer supports v5 fields
SAP OCI 5	<code>sap_oci_v5_outbound_core</code>	use <code>sap_oci_v5_outbound_hierarchy</code> for parent-child data, or <code>sap_oci_v5_outbound_secure</code> for strict enforcement
Generic cXML	<code>cxml_outbound_core</code>	add optional enrichments with <code>cxml_outbound_level2</code>
SAP Ariba	<code>ariba_cxml_outbound_core</code>	use <code>ariba_cxml_outbound_level2</code> or strict variant if level-2 fields are mandatory
Coupa	<code>coupa_cxml_outbound_poom_1_2_023</code>	use strict variant for mandatory complete payloads
Oracle Procurement	<code>oracle_cxml_outbound_1b_core</code>	use strict variant when all mapped fields must be enforced

Step 3: Pick Core vs Strict

- `core`: safer for initial onboarding and partial source data quality.
- `strict`: best for mature integrations where every mapped field is required.

Step 4: Validate with Simulator and Trace

Before production:

1. execute simulator request for target protocol
2. verify mapped payload fields in trace
3. confirm procurement-side acceptance

Related Pages

- [Templates Overview](#)
- [OCI and SAP Templates](#)
- [cXML Core Templates](#)
- [Ariba Templates](#)
- [Coupa Templates](#)
- [Oracle Templates](#)

Common Issues

Punchout Documentation · /punchout/troubleshooting/common-issues

Buyer is not recognized

Verify buyer profile activation, protocol, and buyer identifier mapping. Confirm credentials match the incoming request format.

Signature validation fails

Check strict signature setting, shared secret value, and request canonicalization on the caller side.

Request blocked by security checks

Validate source IP against allowlist and confirm hook/return URL policy constraints for the current environment.

Transfer returns replay or conflict

Review idempotency key usage. Reuse only for true retries of the same payload; generate a new key for new business requests.

Cart API returns context errors

Check token availability and context lifecycle state (active vs expired). Ensure the correct token is propagated in each call.

Mapping output is incomplete

Review mapping profile assignment on buyer, required mapping rules, and transformer/value-map configuration.

Support cannot isolate issue quickly

Use trace correlation id and export JSON trace details for deterministic reproduction with the integration partner.

Need structured error definitions

Use the [Error Catalog](#) for category-by-category resolution steps.

Error Catalog

Punchout Documentation · /punchout/troubleshooting/error-catalog

Use this catalog to classify errors quickly and apply the right fix path.

Authentication Errors

BUYER_NOT_FOUND

- Meaning: no active buyer profile matches inbound identity.
- Check: buyer id, protocol, active flag, store scope.
- Fix: correct buyer identifier mapping and activate the profile.

IDENTITY_MISMATCH

- Meaning: protocol identity fields do not match configured buyer identity.
- Check: OCI user identity or cXML sender identity/domain values.
- Fix: align procurement-side identity values with buyer profile settings.

CREDENTIAL_MISMATCH

- Meaning: secret/password/token validation failed.
- Check: configured credentials and requester-side credential payload.
- Fix: rotate and re-enter credentials on both sides.

Security Errors

SIGNATURE_REQUIRED

- Meaning: strict signature mode is enabled and request is unsigned.
- Check: strict signature setting and caller signature support.
- Fix: enable signing on caller side or disable strict mode temporarily in test only.

SIGNATURE_INVALID

- Meaning: request signature does not match expected hash.
- Check: shared secret, canonicalization, payload mutation in transit.
- Fix: revalidate signature generation and secret synchronization.

IP_NOT_ALLOWED

- Meaning: request source IP is outside allowed ranges.
- Check: allowlist values and real source IP.
- Fix: update allowlist with exact trusted egress ranges.

HOOK_URL_REJECTED

- Meaning: callback/return URL failed policy validation.
- Check: URL host, scheme, and configured restrictions.
- Fix: register approved URL and avoid untrusted redirects.

Transfer and Mapping Errors

MAPPING_REQUIRED_FIELD_MISSING

- Meaning: one or more required mapping fields are empty.
- Check: mapping rule required flags and source field availability.
- Fix: provide source data or add safe defaults where appropriate.

TEMPLATE_PROTOCOL_MISMATCH

- Meaning: template protocol/direction conflicts with profile settings.
- Check: mapping profile protocol, direction, selected template.
- Fix: choose compatible template or adjust profile configuration.

TRANSFER_PAYLOAD_REJECTED

- Meaning: procurement system rejected payload format or required fields.
- Check: outbound trace payload and platform-specific requirements.
- Fix: update mapping rules based on partner validation feedback.

Idempotency Errors

IDEMPOTENCY_REPLAY

- Meaning: request key already processed with same payload.
- Check: request key reuse policy.
- Fix: no action for true retry; response should be reused.

IDEMPOTENCY_CONFLICT

- Meaning: same key used with different payload.
- Check: key generation logic in caller.
- Fix: generate a new key for each new business request.

Fast Triage Workflow

1. Capture correlation id from trace.
2. Classify error category (auth, security, mapping, idempotency).
3. Validate buyer and mapping profile configuration.
4. Re-run with simulator using corrected inputs.