

---

# FOUR51 CXML PUNCHOUT IMPLEMENTATION GUIDE

---

Version 2.2, April 2017

---

## ABSTRACT

---

This implementation guide details the Four51 functionality for allowing procurement systems such as Ariba, SAP, etc. to punch out to the Four51 e-commerce application. This document specifically focuses on the cXML PunchOut work flow.

---

## TABLE OF CONTENTS

---

Abstract.....	1
Introduction.....	2
Definition of Terms.....	2
Project Timeline.....	2
Overview.....	3
PunchOut Order Flow.....	3
Supported PunchOut Features.....	3
Unsupported PunchOut Features.....	5
Detailed Specifications.....	5
Inbound Setup Request.....	6
cXML Setup Request Sample.....	6
Outbound Order Message.....	8
cXML Order Message Sample.....	8
Inbound Order Request.....	10
cXML Order Request Sample.....	10
Troubleshooting and Support.....	12
Seller Support.....	12
Buyer Support.....	12

Error Codes.....	12
References.....	14

---

## INTRODUCTION

---

### DEFINITION OF TERMS

---

**PunchOut** is the name that Ariba (a leading procurement system) has given its round-trip order process. It's now used as a general term for a type of integration between a procurement system and a vendor's e-commerce site (Four51). **OCI** is SAP's version of this integration that is slightly different in behavior, but accomplishes the same goal.

For purposes of this guide, the term '**PunchOut**' refers to cXML PunchOut work flows. A separate document details the OCI work flow.

The **Seller** is typically Four51's direct customer, and either sells or sells/supplies the items ordered through a PunchOut. The **Buyer** is the end customer of the **Seller**, and uses a procurement system to purchase items on Four51.

### PROJECT TIMELINE

---

A Four51 specialist will work with you throughout your PunchOut implementation. Average project time is 3-6 weeks, but may vary significantly based on factors such as: availability of the Seller resources, Buyer resources, Four51 Buyer site build out, and more. Requests for additional services may also increase project cost and timeline. Steps for a standard PunchOut implementation are:

1. Seller and their Buyer agree to implement PunchOut
2. Seller contacts Four51 to begin implementation
3. Kick-off call between all three parties to discuss the project at a high level
4. Four51 Buyer site is constructed and a template user is created

5. Buyer completes the setup form and returns to Four51
6. Four51 configures the test environment and provides testing URLs
7. Buyer tests connection and event sequence
8. Once testing is complete, Four51 provides production URLs and Buyer goes live

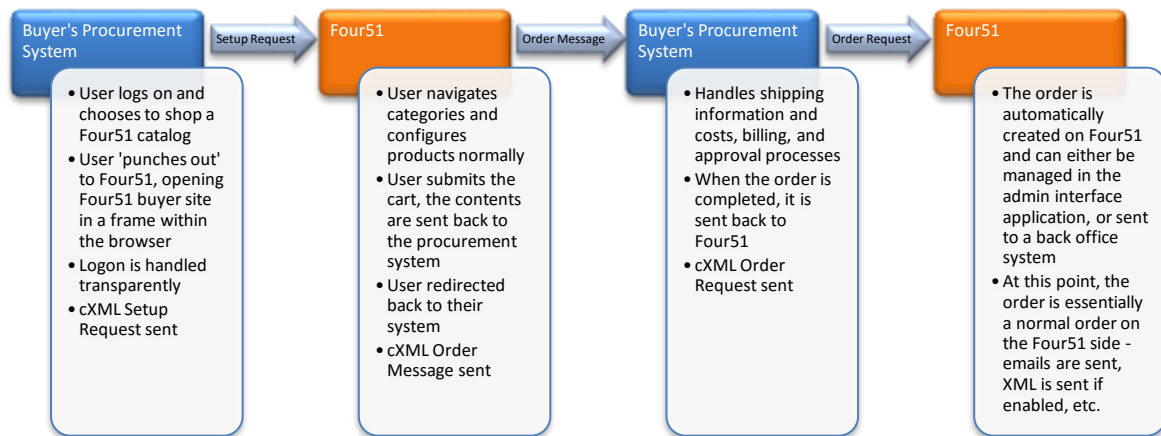
## OVERVIEW

---

### PUNCHOUT ORDER FLOW

---

At a high level, the Four51 PunchOut flow involves three cXML transactions between the Four51 system and the buyer's procurement platform, and will look like this:



### SUPPORTED PUNCHOUT FEATURES

---

- cXML-based round-trips
- Base-64 or URL encoding of the outbound order message
- Custom mapping of incoming and outgoing transactions
- Create, Edit, and Inspect sessions
- UNSPSC commodity codes
- PunchOut Index Catalog (PunchOut directly to a product or category)

- Individual buyer user order history on Four51
- Credit card payment method
- Assign users to group(s)

## UNSUPPORTED PUNCHOUT FEATURES

---

- Setup Request source sessions
- Change orders
- Persistent shopping carts
- Purchase order processing, invoicing, ship notices, and other transactions
- EDI, or anything besides XML-based (cXML) receipt of purchase orders
- Kits

## DETAILED SPECIFICATIONS

---

Four51 is designed to support cXML PunchOut transactions. Four51 can configure each transaction differently, depending on the Buyer's requirements. The transactions are:

- Inbound Setup Request – Used to initiate a user session with Four51, cXML document contained in the body of an HTTP post
- Outbound Order Message – Used to submit the contents of a shopping cart to Buyer's procurement system, cXML document contained in the body of an HTTP post
- Inbound Order Request – Used to submit the final purchase order from Buyer's procurement system to Four51, cXML document contained in the body of an HTTP post

## INBOUND SETUP REQUEST

---

---

### CXML SETUP REQUEST SAMPLE

---

Four51 Element	xPath to Value in Document
Identity	cXML/Header/From/Credential/Identity
SharedSecret	cXML/Header/Sender/Credential/SharedSecret
PunchOut BuyerCookie	cXML/Request/PunchOutSetupRequest/BuyerCookie
Group	cXML/Request/PunchOutSetupRequest/Extrinsic[@name='Group']
Username	cXML/Request/PunchOutSetupRequest/Extrinsic[@name='UniqueName']
First Name	cXML/Request/PunchOutSetupRequest/Extrinsic[@name='UserEmail']
Last Name	cXML/Request/PunchOutSetupRequest/Extrinsic[@name='UserEmail']
Email	cXML/Request/PunchOutSetupRequest/Extrinsic[@name='UserEmail']
Browser Form Post URL	cXML/Request/PunchOutSetupRequest/BrowserFormPost/URL

```
<!DOCTYPE cXML SYSTEM "http://xml.cxm.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US" payloadID="23423424232@customer.com" timestamp="2008-06-18T10:38:54-05:00">
  <Header>
    <From>
      <Credential domain="NetworkId">
        <Identity>AN01000002779-T</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>1234567890</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="NetworkId">
        <Identity>CustomerSender</Identity>
        <SharedSecret>sSecret!</SharedSecret>
      </Credential>
      <UserAgent>SenderAgent</UserAgent>
    </Sender>
  </Header>
  <Request>
    <PunchOutSetupRequest>
      <BuyerCookie>
        <Group>
          <UniqueName>
            <UserEmail>
              <UserEmail>
                <BrowserFormPost>
                  <URL>
                </BrowserFormPost>
              </UserEmail>
            </UniqueName>
          </Group>
        </BuyerCookie>
      </PunchOutSetupRequest>
    </Request>
  </Request>
</cXML>
```

```
</Sender>
</Header>
<Request>
  <PunchOutSetupRequest operation="create">
    <BuyerCookie>111222333444555666777</BuyerCookie>
    <Extrinsic name="Group">Marketing</Extrinsic>
    <Extrinsic name="User">Test</Extrinsic>
    <Extrinsic name="UserEmail">demomail@four51.com</Extrinsic>
    <Extrinsic name="UniqueName">four51testa4wf4334rr</Extrinsic>
    <BrowserFormPost>
      <URL>http://customer.com/ingestpoom.aspx</URL>
    </BrowserFormPost>
    <SupplierSetup>
      <URL>http://www.customer.com/punchoutsubmit/submit.aspx</URL>
    </SupplierSetup>
  </PunchOutSetupRequest>
</Request>
</cXML>
```

## OUTBOUND ORDER MESSAGE

---

---

### CXML ORDER MESSAGE SAMPLE

---

```
<?xml version="1.0"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.005/cXML.dtd"[]>
<cXML timestamp="2008-06-23T13:52:03" payloadID="7a6c960d-b287-4c04-84fd-d6bd12564ff6" xml:lang="en-US">
  <Header>
    <From>
      <Credential domain="DUNS">
        <Identity>8675310</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>8675309</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="DUNS">
        <Identity>134470637</Identity>
      </Credential>
      <UserAgent>Four51 WildFire</UserAgent>
    </Sender>
  </Header>
  <Message>
    <PunchOutOrderMessage>
      <BuyerCookie>DdF7dbfibGII0V1x0.2416756344220453216</BuyerCookie>
      <PunchOutOrderMessageHeader operationAllowed="edit">
        <Total>
          <Money currency="USD">379.00</Money>
        </Total>
      </PunchOutOrderMessageHeader>
      <ItemIn quantity="1">
        <ItemID>
          <SupplierPartID>731791</SupplierPartID>
          <SupplierPartAuxiliaryID>pov2-IVbzuc8E42Ax4RnrXAt02w-e</SupplierPartAuxiliaryID>
        </ItemID>
        <ItemDetail>
          <UnitPrice>
            <Money currency="USD">379.000000</Money>
          </UnitPrice>
          <Description xml:lang="en-US">Sleeping Bag</Description>
          <UnitOfMeasure>EA</UnitOfMeasure>
          <Classification domain="UNSPSC">234992835</Classification>
          <Extrinsic name="ProductSpecs">
            <Extrinsic name="Color">Blue</Extrinsic>
            <Extrinsic name="Size">Small</Extrinsic>
          </Extrinsic>
        </ItemDetail>
      </ItemIn>
    </PunchOutOrderMessage>
  </Message>
```

</cXML>

## INBOUND ORDER REQUEST

---

---

### CXML ORDER REQUEST SAMPLE

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.018/cXML.dtd"[]>
<cXML timestamp="2008-06-24T11:14:09-07:00" payloadID="1214331248548-4660830611117593686@216.109.111.30">
  <Header>
    <From>
      <Credential domain="NetworkID">
        <Identity>AN01000002779-T</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="NetworkID">
        <Identity>AN01000057962-T</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="AribaNetworkUserId">
        <Identity>ariba-network-catalog-tester@ariba.com</Identity>
        <SharedSecret>four51four51</SharedSecret>
      </Credential>
      <UserAgent>CatalogTester</UserAgent>
    </Sender>
  </Header>
  <Request>
    <OrderRequest>
      <OrderRequestHeader type="new" orderID="CTPO302303" orderDate="2008-06-24T11:14:09-07:00" orderVersion="1">
        <Total>
          <Money currency="USD">379.00000000</Money>
        </Total>
        <ShipTo>
          <Address addressID="Primary Ship To Address">
            <Name xml:lang="en-US">XYZ Corporation</Name>
            <PostalAddress name="default">
              <DeliverTo>First Name</DeliverTo>
              <DeliverTo>Last Name</DeliverTo>
              <Street>1234 Ship To Address</Street>
              <Street>2nd Line Address</Street>
              <City>Ship To City</City>
              <State>MN</State>
              <PostalCode>Zip Code</PostalCode>
              <Country isoCountryCode="US">United States</Country>
            </PostalAddress>
            <Phone name="work">
              <TelephoneNumber>
                <CountryCode isoCountryCode="US">1</CountryCode>
                <AreaOrCityCode>800</AreaOrCityCode>
                <Number>555-1234</Number>
              </TelephoneNumber>
            </Phone>
          </Address>
        </ShipTo>
      </OrderRequestHeader>
    </OrderRequest>
  </Request>
</cXML>
```

```

</Address>
</ShipTo>
<BillTo>
  <Address addressID="Primary Bill To Address">
    <Name xml:lang="en-US">XYZ Corporation</Name>
    <PostalAddress name="default">
      <DeliverTo>First Name</DeliverTo>
      <DeliverTo>Last Name</DeliverTo>
      <Street>1234 Bill To Address</Street>
      <Street>2nd Line Address</Street>
      <City>Ship To City</City>
      <State>MN</State>
      <PostalCode>Zip Code</PostalCode>
      <Country isoCountryCode="US">United States</Country>
    </PostalAddress>
    <Phone name="work">
      <TelephoneNumber>
        <CountryCode isoCountryCode="US">1</CountryCode>
        <AreaOrCityCode>800</AreaOrCityCode>
        <Number>555-7891</Number>
      </TelephoneNumber>
    </Phone>
  </Address>
</BillTo>
<Shipping>
  <Description xml:lang="en-US">My Shipping Co</Description>
</Shipping>
<Comments xml:lang="en-US">Test Comments</Comments>
<Payment>
  <PCard name="Visa" number="4111111111111111" expiration="2011-03-12" />
</Payment>
</OrderRequestHeader>
<ItemOut lineNumber="1" quantity="1.00" requestedDeliveryDate="2005-06-15T00:00:00">
  <ItemID>
    <SupplierPartID>731791</SupplierPartID>
    <SupplierPartAuxiliaryID>v3IVbzuc8E42Ax4RnrXAt02w-e-e</SupplierPartAuxiliaryID>
  </ItemID>
  <ItemDetail>
    <UnitPrice>
      <Money currency="USD">211.000000</Money>
    </UnitPrice>
    <Description xml:lang="en-US">Marmot Helium +15 Sleeping Bag - Long</Description>
    <UnitOfMeasure>EA</UnitOfMeasure>
    <Classification domain="UNSPSC">UNSPC 234992835</Classification>
    <Extrinsic name="CostCenter">239840 - Test Cost Center</Extrinsic>
    <ManufacturerPartID>
    </ManufacturerPartID>
    <ManufacturerName>
    </ManufacturerName>
    <URL>http://test.four51.com/ui/hEiaw5beNNzzRA0CkAB-piw-e-e/advtravelcxml/PunchoutOrder.hcf</URL>
  </ItemDetail>
</ItemOut>
</OrderRequest>
</Request>
</cXML>

```

## TROUBLESHOOTING AND SUPPORT

---

### SELLER SUPPORT

---

Questions can be submitted to Four51 by submitting a case in the Four51 Customer Resource Center. This can be accessed through the knowledgebase as follows:

- Logon to your admin interface on Four51 (<http://www.four51.com/ui/logon.aspx>)
- Select 'Help' in the upper right corner
- Click on 'Find Answers'

Please check the knowledge base first, as your question may have already been posted there. If you need additional help, the please select one of the following links:

- Click on the 'Submit A Case Here' link in the lower right-hand side of the panel; or
- Click on 'Click Here To Submit A Case' on the home page of the knowledgebase

### BUYER SUPPORT

---

Please route all questions and issues through your Seller. Communication through e-mail or phone is possible, but this allows the Seller to be aware of all issues, and establishes a standard support process.

### ERROR CODES

---

All successful Setup Request or Order Request transactions will have a response code "200 OK". All unhandled errors will have response code 500 and exception details as the response text. All handled errors will have response code 406 and response text:

### Setup Request Errors

Response Text	Cause
<i>template name not found in default location</i>	The configuration does not exist on Four51, or an incorrect URL was used
<i>Missing required element: &lt;Element&gt;</i>	A required element is missing from the request
<i>Missing required form element: &lt;Element&gt;</i>	A required element is missing from the request
<i>Validation element has incorrect value: &lt;Validation value&gt;</i>	A validation element is expecting a certain value that is not being passed
<i>template user missing</i>	Configuration does not reference a template user
<i>Couldn't find template user</i>	The template user referenced in the configuration does not exist on Four51
<i>user name in use. Couldn't create new user.</i>	The username passed is already in use on Four51 so a new user cannot be created - the password being passed could also be incorrect for the user. May also be caused by the user not having the 'Punchout User' permission.
<i>Couldn't find Four51User.</i>	Generic error with user logon
<i>custom user field not found. &lt;Field Name&gt;</i>	The configuration is attempting to pass a value to a Custom User Field but cannot find a field with that name on Four51
<i>File custom user fields are not supported. &lt;Field Name&gt;</i>	Four51 does not support passing Custom User Fields of the 'file' type

### Order Request Errors

<i>Missing required element: &lt;Element&gt;</i>	A required element is missing from the request
<i>Validation element has incorrect value: &lt;Validation value&gt;</i>	A validation element is expecting a certain value that is not being passed
<i>the order could not be created on Four51 because there is already an order with the order id.</i>	The buyer company is configured to enforce unique order IDs, and there is already an order with the same ID - this is commonly seen when a failed order is in the unsubmitted status but has an order ID

<i>this variant id has expired, please place the order again.</i>	This error usually occurs when unexpected text is sent back as the Four51 part ID. The part ID must be blank for non-catalog orders, or start with 'v3' and be a GUID that represents a variant on Four51.
<i>Order contains a deleted variant.</i>	The part ID is valid, but does not reference an existing variant on Four51. The variant was most likely deleted by the user that placed the order. This typically happens in long approval processes, or when the final Order Request is never sent to Four51
<i>an address with the name &lt;AddressName&gt; exists. The admin company is set up to enforce that address names are unique.</i>	The buyer company is configured to enforce unique address names, and a new address is being created through the order that has a conflicting name
<i>Malformed quantity</i>	Quantity passed must be a positive integer value
<i>Exception has been thrown by the target of an invocation</i>	There is an error with the SupplierPartAuxiliaryID being passed. This is most likely caused by using SPAIDs from a different environment (test vs. production).
<i>Override unit price is enabled, but there is a malformed unit price value.</i>	Unit price passed must be a positive value in the correct format
<i>Malformed punchout line number</i>	Line number passed must be a positive integer value
<i>invalid date &lt;DateNeeded&gt;</i>	Credit card information – self explanatory
<i>Invalid Credit Card Type</i>	Credit card information – self explanatory
<i>Invalid cc expiration date</i>	Credit card information – self explanatory
<i>The credit card expiration date you entered has expired.</i>	Credit card information – self explanatory
<i>Invalid cc number</i>	Credit card information – self explanatory
<i>Failed to contact remote server to calculate tax</i>	This will occur if: A) The tax service is down or not functioning or B) A user is not assigned to the products/variants being submitted on the order.

## REFERENCES

---

- **cXML 1.2.021 Specification** – Chapter 4 of the cXML User’s Guide details the PunchOut Specification and can be found at <http://www.cxml.org>

- **Ariba Supplier Network** – Information about joining the Ariba Supplier Network can be found at <http://supplier.ariba.com>
- **Perfect Commerce Supplier Network** – Information about joining the Perfect Commerce Supplier Network can be found at <http://www.perfect.com/suppliers/suppliers.html>
- **Four51 PunchOut – OCI Implementation Guide** – This document details the similar, yet different, OCI work flow that Four51 supports