



# Ariba PunchOut Implementation Guide

DECEMBER 2000



Copyright 2000 by Ariba, Inc. The information contained in this document is proprietary and confidential to Ariba, Inc. All rights reserved. Patents Pending.

Ariba and the Ariba logo are registered trademarks of Ariba, Inc. Ariba B2B Commerce Platform, Ariba Buyer, Ariba Marketplace, Ariba Dynamic Trade, Ariba Commerce Services Network, buyer.ariba, supplier.ariba, ECTranslator, ECTransport, Walkup UI, Supplier Advisor and Making the Net Work for B2B are trademarks or servicemarks of Ariba, Inc.

All other brand or product names may be trademarks or registered trademarks of their respective companies or organizations.

ALL LICENSES OF ARIBA SOFTWARE PROGRAMS AND RELATED DOCUMENTATION (“PROGRAMS”) ARE SUBJECT TO ANY EXPORT LAWS, REGULATIONS ORDERS OR OTHER RESTRICTIONS IMPOSED BY THE UNITED STATES OF AMERICA OR BY ANY OTHER GOVERNMENT ENTITY ON THE PROGRAMS OR INFORMATION RELATING THERETO. A LICENSEE OF ANY PROGRAM WILL NOT IMPORT, EXPORT, OR ALLOW THE EXPORT OR REEXPORT, DIRECTLY OR INDIRECTLY, OF THE PROGRAM (OR TECHNICAL DATA OR OTHER INFORMATION RELATED THERETO) OR ANY DIRECT PRODUCT THEREOF, TO ANY COUNTRY TO WHICH SUCH IMPORT, EXPORT, OR REEXPORT IS RESTRICTED OR PROHIBITED, OR AS TO WHICH SUCH GOVERNMENT OR ANY AGENCY THEREOF REQUIRES ANY EXPORT LICENSE OR OTHER GOVERNMENTAL APPROVAL AT THE TIME OF IMPORT, EXPORT OR REEXPORT, WITHOUT FIRST OBTAINING SUCH APPROVAL.

---

# Table of Contents

<b>Preface</b> .....	<b>v</b>
Audience and Prerequisites.....	v
Related Documentation.....	v
Typography.....	vi
Ariba Technical Support.....	vi
<b>Chapter 1</b>	
<b>Supplier PunchOut Implementation</b> .....	<b>1</b>
Implementation Methodology.....	1
Planning .....	2
Analysis of Present and Future State.....	2
Outsourcing Versus Internal Development .....	3
Design.....	6
Supplier Specifics .....	6
Buyer’s Business Requirements .....	7
Buyer Specifications .....	8
PunchOut Process Flow.....	9
Development.....	9
Message Flow .....	10
Specifying the PunchOut URL .....	13
Extrinsic and Supplier Cookies .....	15
Multiple From Credentials in Marketplace Transactions .....	15
Support for Profile Transaction.....	16
Testing.....	17
Self-Testing on Ariba CSN .....	17
Testing with Ariba Global Solutions .....	18
Testing with the Client.....	18
Deployment.....	21

<b>Chapter 2</b>	
<b>Expediting Development of a PunchOut Site . . . . .</b>	<b>23</b>
cXML Messaging . . . . .	24
<b>Chapter 3</b>	
<b>Retrofitting an Existing Website . . . . .</b>	<b>25</b>
Leveraging the Existing Site. . . . .	25
<b>Chapter 4</b>	
<b>Integrating Service Organizations . . . . .</b>	<b>27</b>
PunchOut for Services . . . . .	27
Milestones. . . . .	28
<b>Chapter 5</b>	
<b>Creating a PunchOut Deployment Guide. . . . .</b>	<b>31</b>
Outline . . . . .	31
Connectivity Overview . . . . .	32
Authentication and Identification . . . . .	32
Required Extrinsic. . . . .	33
Content Requirements/Specification . . . . .	33
Address Information . . . . .	33
Accounting Structure . . . . .	34
Commodity Coding. . . . .	35
Transactions Supported. . . . .	35
Change/Cancel Orders. . . . .	36
Ad Hoc/Non-Catalog Line Items within an Order . . . . .	36
Internal Catalog Orders . . . . .	36
Copied Requisitions. . . . .	36
Order Exceptions . . . . .	36
Third Party Suppliers. . . . .	37
Quote Split into Multiple Orders. . . . .	37

---

<b>Chapter 6</b>	
<b>Sample Code</b> . . . . .	<b>39</b>
cXML Index File . . . . .	39
PunchOut Setup Request Document . . . . .	40
Key Elements . . . . .	40
Sample . . . . .	42
PunchOut Setup Response Document . . . . .	43
Key Elements . . . . .	43
Sample . . . . .	44
PunchOut Order Message Document . . . . .	44
Key Elements . . . . .	44
Sample . . . . .	46
Order Request Document . . . . .	47
Key Elements . . . . .	47
Sample . . . . .	48
Order Response Document . . . . .	51
Sample . . . . .	51
Rapid Site Deployment . . . . .	52
File receivePunchoutSetupRequest.asp . . . . .	52
File resolveXML.asp . . . . .	53
File TCcXMLFormatDTime.asp . . . . .	56
<b>Index</b> . . . . .	<b>59</b>



---

# Preface

This document describes the process for successfully integrating suppliers into a buying organization's Ariba Buyer procurement environment and Ariba Marketplace, Network Edition. The focus is on the various steps: planning, design, development, testing, and deployment of a production-ready PunchOut site.

## Audience and Prerequisites

---

This document is intended for suppliers that are creating a PunchOut enabled Website. The audience for this methodology includes suppliers, system integrators, and supplier enablement service providers.

Readers should have a working knowledge of e-commerce concepts, XML, and the HTTP Web communication standard.

## Related Documentation

---

### *cXML User's Guide*

Describes how to use cXML (commerce eXtensible Markup Language) for communication of data related to electronic commerce.

### *Ariba Catalog Format Reference*

Describes available catalog features and the syntax of CIF 3.0 and cXML catalogs.

### *Ariba CSN Supplier's Guide*

Describes Ariba Commerce Services Network (CSN) in detail, including a description of how it works and instructions on registering and setting up supplier accounts.

## Typography

cXML elements and attributes are denoted with a monotype font. cXML element and attribute names are case-sensitive. Both are a combination of lower and uppercase, with elements beginning with an uppercase letter, and attributes beginning with a lowercase letter. For example, MyElement is a cXML element, and myAttribute is a cXML attribute.

The following table describes other typographic conventions used in this book:

Typeface or Symbol	Meaning	Example
<code>&lt;AaBbCc123&gt;</code>	Text you need to change is italicized, and appears between angle brackets.	<code>http://&lt;yourServer&gt;:&lt;HTTPServerPort&gt;/inspector</code>
<b>AaBbCc123</b>	The names of user interface controls, menus, and menu items.	Choose <b>Edit</b> from the <b>File</b> menu.
AaBbCc123	Files and directory names, parameters, fields in CSV files, command lines, and code examples.	There is one line in ReportMeta.csv for each report in the system.
<i>AaBbCc123</i>	The names of books.	For more information, see <i>Ariba Buyer Configuration Overview</i> .

## Ariba Technical Support

For assistance with Ariba Commerce Services Network and Ariba Marketplace Network Edition, technical support is available by phone and e-mail:

Hours of Operation: 24 hours a day, Monday through Friday (excluding U.S. holidays)

Phone: (650) 930-6400 or 1-888-89ARIBA

North America: 1-888-89ARIBA

Europe, Middle East, Africa: +31 (0)23 5685777

Australia: 1800 993346

New Zealand: 0800 443167

Singapore: 800 6161703

Hong Kong: 800 900163

E-mail: network\_support@ariba.com or marketned\_support@ariba.com



---

# Chapter 1

## Supplier PunchOut Implementation

To the Ariba Buyer user, the key difference between PunchOut and local catalogs is what they see in the User Interface (UI). For both, the UI displays a *catalog hierarchy* enabling users to choose items, listed as *line items* on requisitions. Functionally, the difference between the two is that PunchOut catalogs require hosting by you, the supplier, while local catalogs are completely hosted by Ariba CSN.

For local catalogs, CIF files typically populate both the catalog hierarchy and requisition line items. With PunchOut however, populating the catalog hierarchy and requisition line items use separate vehicles. First, you provide a PunchOut index file that is loaded into Ariba Buyer that populates the catalog hierarchy. Second, the submitting of a cXML message to Ariba Buyer from the check-out page of your PunchOut site populates requisitions with line-items. Ariba Buyer does not require any customization to be able to handle these cXML messages, tagged with the cXML element `PunchOutOrderMessage`.

### Implementation Methodology

---

The supplier PunchOut implementation process spans from initial evaluation of a supplier's system to being *Ariba Ready*. You obtain certification when the PunchOut site is ready to exchange documents with an Ariba customer.

The steps in creating a PunchOut site follow, in this order:

- Planning
- Design
- Development
- Testing
- Deployment

The information in this document should be used as a guide to assist you in implementing a PunchOut site and for defining a process to avoid potential pitfalls that can cause delays in production. Many of the tasks described in this document are completed only once. By following the methodology suggested in this document, you have the advantage of creating an initial solution that is scalable for future PunchOut implementations.

## Planning

---

The Planning phase should include the development of a high-level vision of the integration as well as a rough cut at the project plan and resources required.

### Analysis of Present and Future State

Before developing a PunchOut site, perform an in-depth analysis of your current system and what is needed to support the “new” process. A few things to note:

For the current state:

- Do you have an existing e-commerce site deployed?
- Does your site have XML integration enabled?
- Do you currently receive orders electronically through XML or EDI?
- How is integration performed with back-end systems?
- Do you currently use the customer’s control data (e.g., ship location ID’s) to identify pricing and shipping of orders?
- Does your site currently have the ability to deliver custom pricing or content based on the customer ID?

For the future state:

- How will the process flow from time of shopping through order placement and fulfillment?
- Can a single product line be selected for an initial pilot of PunchOut integration?
- Will you need to interface with multiple XML-based procurement applications?
- How should your concept of a “shopping cart” be modeled in requisition line items?

## Outsourcing Versus Internal Development

Next, you should review outsourcing versus building the integration yourself. The Ariba Supplier Advisor, available at [supplier.ariba.com](http://supplier.ariba.com), will help you in making this decision and offer suggestions for partners to assist in the technical integration. Some key factors to take into account:

- Does your existing technical staff have the expertise to implement a PunchOut site?
- Do you have an existing Web infrastructure that can be leveraged relative to Ariba's protocols?
- Do you have an approved budget for e-commerce initiatives, in particular, Ariba Supplier Network enablement?
- Has you evaluated the process flow with PunchOut to verify that it fits into your business model for offering service procurement?

### ***Key Participants***

The Ariba supplier integration process is a collaborative effort that leverages the skills of Ariba, client, and supplier resources. The key participants typically involved in the process are as follows:

#### **Supplier Technical Developer**

You are strongly encouraged to identify a qualified team member to assume the role of Technical Developer for the PunchOut site. This person acts as the primary owner of the supplier PunchOut process and assumes responsibility for a number of tasks. The following is a role description for the Supplier Technical Developer:

- Main recipient of PunchOut documentation available from Ariba online from the following Websites:

<http://supplier.ariba.com>

<http://cxml.org/home/>

<http://xml.cxml.org/>

- Responsible for familiarity with Ariba methodology; develops or possesses a strong working knowledge of catalog content formats including:
  - CIF
  - cXML (commerce eXtensible Markup Language)

- PunchOut
- Main point of contact when testing the PunchOut site with Ariba and the client
- Manages the Ariba Supplier account for all technical PunchOut related matters

### **Supplier Integration Manager**

The Supplier Integration Manager is the main point of contact for non-technical issues relating to PunchOut and enabling a site. The role description for this person includes:

- Verifying the Dun and Bradstreet (D&B) number for your company,
- To find out or obtain the D&B number, visit: <http://www.dnb.com/dunsno/whereduns.htm>. Registering for a D-U-N-S® number is free, but it can take up to three weeks to process.
- Managing the relationship between PunchOut customer and supplier; resolving issues such as:
  - Identifying targeted commodities for PunchOut
  - Supplying commodity codes for PunchOut products to the buyer
  - Defining the invoice process with the buyer
- Creating a project plan and time line for implementation
- Determining additional resource requirements and making appropriate assignments

### **Ariba Supplier and Content Solutions Consultants**

Ariba Consultants can provide targeted support on specific topics such as Ariba Commerce Services Network (CSN) registration and functionality, catalog formats, testing approaches, etc. Ariba CSN supports suppliers at any level of technical sophistication.

### ***Determining the Level of Support***

Your requirements, order volume, the amount of automation desired, and your level of expertise determine the level of support needed. Answering the following questions can help determine the level of support needed.

**Do you have a solid understanding of XML and cXML?**

XML (eXtensible Markup Language) provides the building blocks of any cXML document. cXML documents provide a way for buyers, suppliers, and Ariba CSN to communicate with each other. cXML documents are constructed based on Document Type Definition (DTD) files which are used to define a content model for a cXML document. A DTD includes the specifications for the elements allowed, their order, and the data types of attributes.

**Do you currently have any transactive, XML-enabled Web-based e-commerce applications?**

Suppliers implementing PunchOut must have a fundamental understanding of how to create, parse, query, receive, and transmit XML data to and from a remote source. The basic tool to process XML messages is an XML parser. Various tools available free of charge to familiarize the technical team with XML and cXML. You can find trial copies of the schema and instancing tools XML Authority and XML instance at <http://www.extensibility.com/products/index.htm>. Several third party tools and services exist to accelerate the implementation of an XML enabled infrastructure. Contact Ariba CSN Support at [network\\_support@ariba.com](mailto:network_support@ariba.com) for details.

**Do you currently have a catalog index file?**

A catalog index file is a cXML or Catalog Information Format (CIF) document you create that the buyer loads into Ariba Buyer to create entries in the catalog hierarchy. The file defines how you and your products appear in the Ariba Buyer catalog UI. For a CIF catalog, a UNSPSC code in the index file is mapped to an internal commodity code in Ariba Buyer. The objects residing in the buyer system must be pre-configured for you and the commodities being imported.

A catalog entry is created when the index file has been imported and approved. When the user selects the catalog entry, Ariba Buyer formats and sends a PunchOut Setup Request document to Ariba CSN, which initiates the session. A Web browser then display the PunchOut catalog on your Website. See “cXML Index File” on page 39 for an example of a cXML index file.

## Design

---

The overall design of a PunchOut site and how it will integrate with a supplier's system is critical to the implementation process.

Since certain catalog formats are better for certain commodities and business rules, you must understand them and discuss them with the client to understand buyer requirements. You must address the following areas to complete this analysis:

- Supplier specifics
- Buyer's business requirements
- Buyer specifications
- PunchOut process flow

### Supplier Specifics

There is flexibility in the appearance and functionality of your PunchOut site derived from supplier specific aspects of your Punchout design, listed as follows:

- Branding
- Publishing the catalog
- Communicating with other Ariba Network suppliers
- Ariba Ready

Several factors, such as schedule, budget, and appearance will influence how you approach some of the supplier specifics and you should review all of them as part of your design process.

#### ***Branding***

PunchOut catalogs enable the supplier to differentiate themselves from other suppliers. Rather than simply displaying product or pricing details, Ariba Buyer displays a hyperlink taking users directly to custom product pages on your PunchOut site. Users can browse product options, specify configurations, and select delivery methods. Features such as user authentication and personalization, can be made available to buying organizations. Supplier's are encouraged to personalize their PunchOut site and leverage the "look and feel" of any existing customer sites they may already have. Ariba does require certain features in the PunchOut site, such as links to exit the current shopping session and to access external sites.

## ***Publishing the catalog***

PunchOut catalogs are published on Ariba CSN like other catalog types. The supplier can make the catalog public and available to all buyers, or private and available only to specific buyers. Buyers select catalogs to view based on their description and on details about the supplier. Publishing a catalog to Ariba CSN is the quickest way let buyers know that a new supplier has come online.

## ***Communication with other Ariba Network Suppliers***

Ariba hosts two supplier forums each year for the exchange of information about supplier e-commerce requirements and to deliver updates on the state of the industry. These summits provide an excellent educational and networking forum, and deliver insight into supplier challenges and opportunities in e-commerce. These meetings are limited to ASL (Ariba Supplier Link) members.

Ariba also hosts two *Customer Advisory Council* meetings each year. These events are open to all Ariba customers, including non-ASL members. The Customer Advisory Council is another great medium for interacting with both buyers and suppliers, as well as learning about the latest state of Ariba products.

## ***Ariba Ready***

*Ariba Ready* is a service offered at the [supplier.ariba.com](http://supplier.ariba.com) website. Once the PunchOut site is enabled and tested thoroughly with the PunchOut Test Tool, you should apply for Ariba Ready, where the PunchOut site is placed in a queue to be tested by the Ariba Ready team. Once testing is complete, the site is passed to an Ariba partner for script testing. When the site passes the scripts, the supplier receives an Ariba Ready logo. This may be placed on the site to notify potential buyers that the supplier has met Ariba's PunchOut requirements, expediting the addition of buyers to the site. Once the supplier account is active, obtain information about Ariba Ready at [supplier.ariba.com](http://supplier.ariba.com). To access Ariba Ready, login to your supplier account and click the **Become Ariba Ready** link under **Quick Links**.

## **Buyer's Business Requirements**

Meet with your customers to determine their business requirements. You should:

- Complete a gap analysis of current and future procurement practices
- Determine content-specific requirements by commodity
- Identify any reporting considerations or requirements
- Assign ownership across the team to resolve open issues

Define with the buyer which products are to be used for PunchOut. Determining whether or not PunchOut is the best solution for a supplier is important because development of a PunchOut site can be time-consuming and costly. If you retail only a small number of products for the buyer, a CIF catalog may be more appropriate than PunchOut.

The current state of your system impacts the development time line for creating a PunchOut site. A thorough evaluation your existing system and available resources is imperative to determine a realistic time frame for development through production.

## Buyer Specifications

To build the appropriate functionality in a PunchOut site, review the buyer's specifications. In addition to technical requirements, you should analyze:

- Buyer-specific content and pricing, including national versus regional contracts
- Buyer specific requirements for commodity and Unit of Measure (UOM) encoding
- Buyer specific requirements for international issues such as multi-language and currency

The purpose of this step is to review and develop the processes for addressing the multiple issues that arise when two entities enter as buyer and supplier into a business arrangement. Inspection of the above review points will assist in determining the following.

- How freight, ship method, and taxes will be addressed
- How payment, such as Pcard, invoice, and letter of credit, will be made
- How credit and returns will be processed
- How ad hoc and non-catalog purchase orders will be processed
- How changed and cancelled orders will be processed
- How additional information required for documents and cXML requests, such as cost center, department, requester, and supplier account code, will be handled
- How conflicts with existing sales channels, such as distributors, minority or woman-owned companies, will be handled
- How your mechanism for updating order status on Ariba CSN will be integrated



## PunchOut Process Flow

It is important to document the transaction process flow in and out of your user interface, and to identify which messages need to be coded. Ariba has documentation available to assist in defining the process within an organization. Suppliers registered on Ariba CSN can download the following guides from [supplier.ariba.com](http://supplier.ariba.com) by clicking the **Help** tab:

### *Ariba CSN Supplier's Guide*

Describes Ariba CSN in detail, including a description of how it works and instructions on registering and setting up supplier accounts.

### *Ariba Catalog Format Reference*

Describes available catalog features and the syntax of CIF 3.0 and cXML catalogs.

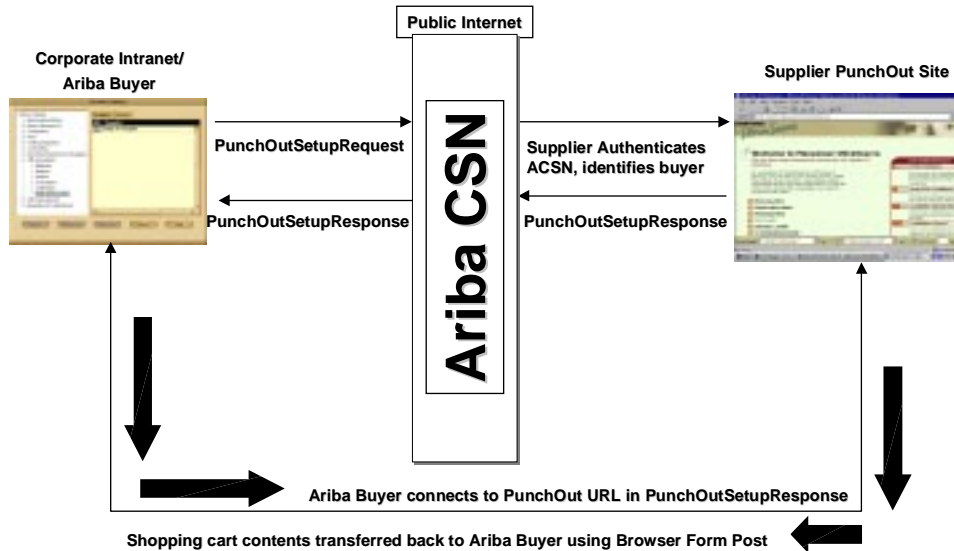
## Development

---

You complete the development phase by enabling each step in the PunchOut process. This process can be described by a message flow, shown in the diagram below, followed by the details and the requirements of each step in the process.

## Message Flow

The following transaction flow diagram details the PunchOut process between the PunchOut site and Ariba CSN.



The steps in the message flow are described as follows:

### 1. User login

To create a requisition, the user at a buying organization logs onto Ariba Buyer, creates a requisition, and selects a PunchOut catalog. The supplier's catalog is highlighted, at which point the user executes PunchOut to the supplier PunchOut site. The user may have access store-, aisle-, and product-level PunchOut. If the user has selected an item or aisle from the catalog, they PunchOut to an item or aisle specific page. They also may have the option to PunchOut directly to the supplier's site without first selecting an item, where they can then browse for an item.

### 2. PunchOut Setup Request

A *PunchOut Setup Request* document is sent with an HTTP Post from the buying organization to Ariba CSN, which authenticates and forwards the document through an HTTP Post to the supplier URL specified in the Request. Authentication is performed on a NetworkID (Credential domain), and SharedSecret *sent* in the Sender element.

When a buyer registers on Ariba CSN, the administrator is issued a login ID (usually an e-mail address), and a password. As part of the maintenance process, the buyer may select a SharedSecret or additional password different from the one used for login. A PunchOut Setup Request document contains identification details (domain NetworkID), as well as the supplier's D-U-N-S number. When it is sent to Ariba CSN, the network is able to identify the buyer based on the Identity element in the From root element and populate the Credential domain with the buyer's Network ID. This is the ID the supplier sees in the PunchOut Setup Request and one of two elements used to authenticate the buyer. Each buyer has their own NetworkID.

The supplier targeted by the PunchOut Setup Request is identified by the DUNS number or NetworkID value in the To root element. When Ariba CSN determines who the request is from and where it needs to go, the SharedSecret in the Sender element is populated with the supplier's shared secret value saved at supplier.ariba.com, which is then used to log in to the supplier's site. The supplier never sees the buyer's SharedSecret and does not have to maintain a separate password/login for each user or buyer. Details about the Ariba Buyer user may be captured in Contact and Extrinsic elements. See page 40 for a cXML example of a PunchOut Setup Request document.

In addition to the authentication and identification parameters, the PunchOut Setup Request document contains a BuyerCookie. The BuyerCookie changes between concurrent PunchOut sessions, thereby allowing a supplier to track which screen a user is on during the shopping process. An edit operation on an existing order usually results in a new buyer cookie for that particular session. To be more specific, Ariba procurement applications guarantee that the buyer cookie is unique among all values used by simultaneous PunchOut sessions initiated by that application. The value might, for example, correspond to a session identifier within that application. To link a specific order to a user, use the SupplierPartAuxiliaryID (supplier cookie) element.

### 3. PunchOut authentication

When a supplier site receives the PunchOut Setup Request document, the following takes place:

- Authentication of Ariba CSN is established based on Sender contents,
- The From identification is verified and the user redirected to the supplier's site.
- The supplier can now initiate a session using the buyer's company and user information because the buyer is a certified Ariba CSN user. A link to the new PunchOut session can be established using the returned start page URL.

#### 4. PunchOut Setup Response transmitted

The *PunchOut Setup Response* document is transmitted via cXML through Ariba CSN, confirming that the request was received successfully. The key elements of the response are the Status code which equals '200' if the request was received without error, and the StartPage URL. See page 44 for a cXML example of a PunchOut Setup Response document.

#### 5. Shopping experience

Once in a supplier's Website, the user should be able to navigate easily and without confusion. Selecting an item adds it to a shopping cart or basket in the supplier's site. Upon completion of shopping, the user returns to Ariba Buyer where they can inspect and edit line items procured from the supplier's PunchOut site. This is accomplished by mapping the contents of the shopping cart to cXML and then returning to the user's Ariba Buyer system.

#### 6. PunchOut Order Message created

The *PunchOut Order Message* document sent back to Ariba Buyer contains the contents of the user's shopping cart. Note that the order message is a *quote*, not an actual order. The order for goods does not exist until the quote has been approved within Ariba Buyer. The order message should have the operationAllowed attribute set to "edit", enabling the user to return to the supplier's site and make changes to the order prior to submission for approval on the buyer's side. To alleviate confusion on the Ariba Buyer user side, it is recommended that the checkout process contain the following sequence of 'buttons':

- Add item to basket
- Transfer basket to Ariba
- Submit order for approval to Ariba Buyer

In addition to the above, the checkout process for PunchOut should not require the user to enter credit card information or ship-to address details. This data is maintained on the buyer side. Ship-to information is visible to the supplier in both the PunchOutSetupRequest and later in the order. Upon submitting the quote, the user is returned to Ariba Buyer.

In versions of Ariba Buyer prior to 7.0, because the supplier's site appears in a new browser, there is a final screen that the user sees containing the supplier's purchase requisition number and a button with the text "Close Browser". This returns the user to Ariba Buyer. See page 44 for a cXML example of a PunchOut Order Message document.

## 7. Requisition submitted for approval

Ariba Buyer submits the quote for approval and routes to workflow. At the present time, Ariba Buyer does not update the supplier on the progress of an order after it has been submitted for approval. If a manager on a workflow chain denies the order, they have the *option* of using PunchOut to the supplier's site to remove line items or to delete the requisition. However, this is not required. Ariba recommends that the supplier reach an agreement with the buyer as to how a *cancelled requisition* will be handled in the production environment.

## 8. Order Request created

An *Order Request* document is created upon approval and transmitted to the supplier through Ariba CSN. This document contains the purchase order details required by the supplier for processing. Orders can be viewed by the supplier in the supplier account inbox on [supplier.ariba.com](http://supplier.ariba.com). Within the supplier account, the status of an order can be changed or updated by the supplier. See page 47 for a cXML example of an Order Request document.

## 9. Order Response generated

The *Order Response* document confirms that the order was received by the supplier and parsed correctly. It is not a commitment to execute a purchase order, only confirmation that it was received and is a valid cXML document. The Response is transmitted via Ariba CSN. The key element is the Status which will be have code equal to "200" and text set to "OK" if there are no errors. See page 48 for a cXML example of an Order Response message.

## Specifying the PunchOut URL

There are two places where you specify your PunchOut site's URL: in your PunchOut index catalog, and in the "Punchout Setup Request URL" field in the Configuration area of your Ariba CSN account.

Ariba CSN uses both URLs.

### ***URLs from Your Index Catalog***

When users punch out, their procurement application adds the URL from your index catalog to the SupplierSetup element in the PunchOut Setup Request document.

From a cXML index catalog:

```
<PunchoutDetail>  
  <Description xml:lang="en-US">Desk Chairs</Description>
```

```
<URL>http://www.workchairs.com/punchout.asp</URL>
<Classification domain="UNSPSC">5136030000</Classification>
</PunchoutDetail>
```

From a CIF index file

```
CIF_I_V3.0
CODEFORMAT: UNSPSC
COMMENTS: This is an example of an punchout catalog item
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item Description,
SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer Name, Supplier URL,
Manufacturer URL, Market Price, PunchOut Enabled
CURRENCY: USD
DATA
762311901,A2C-311F,C-311F,"Desk
Chairs",11116767,,,,,http://www.workchairs.com/punchout.asp,,,t
ENDOFDATA
```

From resulting PunchOut Setup Request document:

```
<SupplierSetup>
<URL>http://www.workchairs.com/punchout.asp</URL>
</SupplierSetup>
```

### ***URL vs. SelectedItem***

Depending on the cXML version used by your PunchOut site, the PunchOut Setup Request might also contain a SelectedItem element specifying the item the user is punching out for:

- cXML 1.0 and ORMS 6.1 do not use SelectedItem. So, the URL in your index catalog is the only way to specify the item to punch out for.
- cXML 1.1 and Ariba Buyer 7.0 use SelectedItem to specify the Item ID of the item to punch out for. Your PunchOut site can ignore the URL within the PunchOut Setup Request, so you can use a bogus URL within your index catalog.

### ***URL Specified on Ariba CSN***

You determine where Ariba CSN forwards the PunchOut Setup Request document by specifying a URL in your Ariba CSN account configuration:

- If you do not specify this URL on Ariba CSN, it routes the PunchOut Setup Request to the URL specified within your index catalog.
- If you specify this URL on Ariba CSN, it routes the PunchOut Setup Request to that URL, not to the URL specified within your index catalog.

It is highly recommended that you enter your PunchOut URL on Ariba CSN, because if you change the location of your PunchOut site, you can easily change this parameter on Ariba CSN, without having to distribute new index catalogs. This parameter enables you to store routing information with the application that needs it (Ariba CSN) instead of distributing it to applications that do not need it. However, the URL returned in the Profile Response from the supplier site always takes precedence.

## Extrinsics and Supplier Cookies

*Extrinsic* data is used to further identify a user to a supplier. The standard extrinsics sent from Ariba Buyer are User and CostCenter. The buying organization will determine the naming and population of all extrinsic elements. Because of this, any additional data the supplier needs to be passed in the PunchOut Setup Request document *must* be coordinated with the buying organization. Heavy reliance on extrinsic data is discouraged, since it makes the scaling of the PunchOut integration to other buying organizations more difficult.

The SupplierPartAuxiliaryID, or “supplier cookie”, allows the supplier to transmit additional data, such as a quote number or another cXML document. The procurement application, Ariba Buyer, passes it back to the supplier in any subsequent PunchOut Setup Request “edit” or “inspect” sessions, and any resulting cXML Order Request. The cookie is often used to associate items in a purchase requisition with the corresponding items in a shopping cart at the supplier’s Website.

## Multiple From Credentials in Marketplace Transactions

If you work with Ariba Marketplace, Network Edition (AM-NE) marketplaces and you have a cXML-enabled Website, you should be aware of a change to cXML documents. All cXML 1.1 documents from marketplaces, such as PunchOut Setup Requests and purchase orders, contain multiple From credentials. A new credential that identifies the member organization will supplement the existing credential that identifies the marketplace host.

You might need to modify your PunchOut Website or order-receiving system to accept multiple From credentials; for example:

```
<From>
  <!-- Acme Enterprises -->
  <Credential domain = "NetworkID">
    <Identity>AN66667777</Identity>
  </Credential>
  <!-- Triton Bank B2B Exchange -->
  <Credential domain = "NetworkID" type="marketplace">
    <Identity>AN223344789</Identity>
  </Credential>
```

```
</From>
<To>
  <!-- Work Chairs, Inc. -->
  <Credential domain = "duns">
    <Identity>942888711</Identity>
    <SharedSecret>abracadabra</SharedSecret>
  </Credential>
</To>
```

The marketplace host's credential is tagged with "type=marketplace". Most suppliers use only the marketplace host's credential, but some suppliers also use the member organization information. The credential you use depends on your business model and your relationship with the marketplace host. Contact the marketplace host if you need more information.

You can include "type=marketplace" in the credential you pass back to AM-NE. For cXML 1.0 documents, AM-NE will continue to include only the marketplace host's credential, and it will leave off the "type=marketplace" attribute. No *Credential* element for the member company is provided to cXML 1.0 suppliers.

## Support for Profile Transaction

Ariba CSN supports the cXML 1.1 Profile transaction. cXML-enabled Websites can use this transaction to look up the URLs for specific Ariba CSN services.

To use the Profile transaction, send a Profile Request document with the following *To* credential:

```
<To>
  <Credential domain="AribaNetworkUserId">
    <Identity>admin@ariba.com</Identity>
  </Credential>
</To>
```

Ariba CSN responds with a list of all Requests that it supports, and the appropriate URLs for those Requests. The information returned changes occasionally; it is recommended that cXML-enabled Websites issue a Profile Request daily.

Ariba CSN sends a ProfileRequest to your cXML-enabled Web site to retrieve a list of your supported services and their URLs. Ariba CSN stores your profile information and uses it for all cXML documents it sends you. Your profile information expires daily, and Ariba CSN updates it with a another ProfileRequest when it needs to send a cXML document to you. For complete information about the Profile transaction, see the *cXML User's Guide*.



---

## Testing

---

The objective of PunchOut testing is to ensure that the PunchOut technology is configured properly and that the supplier's site will effectively communicate with Ariba Buyer. There are three phases of supplier testing:

- Self-testing on Ariba CSN
- Testing with Ariba Global Solutions
- Testing with the client

### Self-Testing on Ariba CSN

Supplier test accounts have a built-in order tester that allows you to check CIF catalogs and order routing. The order tester can be used to send simple purchase orders to the supplier. The tool is useful for debugging a PunchOut-enabled Website and for demonstrating the site to a potential customer.

▼ **To self-test on Ariba CSN:**

1. Log in to the supplier test account.

**Note:** The order tester exists only in test accounts, not in production accounts.

2. Create a PunchOut catalog entry at [supplier.ariba.com](http://supplier.ariba.com).

For specific details on creating this, please see the *Ariba Network Supplier's Guide*.

3. Click "Catalog" to display a list of all available catalogs.

**Note:** Only catalogs in the test account are visible. Catalogs can be tested before or after publishing them, but they must first pass validation. If a "Test" link does not display, the catalog failed validation, or it is not in one of the testable formats.

4. Click the **Test** link next to a catalog.

Ariba CSN displays the contents of the catalog.

5. Choose items by clicking **Add** next to them.

A **Shop** link appears next to PunchOut items. When you click this link, a frame set opens and displays your live Web page for that item.

6. After creating a purchase order, click **Submit** to send it.

Ariba CSN generates a cXML purchase order. You can view the cXML contents and response from Ariba CSN.

7. Examine the orders in your order-receiving system.

This can be done through the Inbox in the [supplier.ariba.com](http://supplier.ariba.com) Website. Ariba CSN displays the contents of the purchase order for troubleshooting.

## Testing with Ariba Global Solutions

- ▼ **To prepare for testing with Ariba Global Solutions, perform the following:**

1. Contact Ariba Support.

Prior to testing with the buyer, contact Ariba support to coordinate the testing effort on the Ariba side and certify that you are ready to receive test orders through the network.

2. Provide D-U-N-S number and PunchOut URL (index file).

The Ariba resource assigned to a supplier account must have the D-U-N-S number and storefront URL to begin the testing process.

In addition to completing these steps, the supplier should have their functional and technical staff available for testing. A series of transactions get the information from the supplier's site, return to the requisition, submit for approval, and approve an order. The time required to perform these transactions depends on the number of transactions tested, the number of commodities included in the testing, and the complexity of the supplier's site.

## Testing with the Client

The final phase of testing is with the Buying Organization. Prior to testing with the customer, however, you should confirm that the buyer has enabled you as a supplier that can publish to their Ariba account. You must make use of the buyer's NetworkID, which is available at [supplier.ariba.com](http://supplier.ariba.com) when viewing the buyer's information. Using an Ariba test account, confirm with the buyer that the catalog content meets their criteria and that orders received into the supplier test account are acceptable by the supplier's order entry system.

A few specific scenarios to run through when testing with the customer are suggested below. The purpose of these illustrations is to give you an idea of how the PunchOut site and Ariba Buyer work together. In addition to the cases listed, it is recommended that you follow the same scenarios used when testing directly with Ariba. For further details, contact Ariba CSN and work with the buyer to help define the testing script.

### **Authentication**

Authentication should be performed through the domain, buyer identity, and shared secret. If authentication is performed any other way, for example with the user id, perhaps with a user-entered password, the site cannot be deployed.

### **Security**

The PunchOut site URL *must* be HTTPS (secure). HTTP may be used for testing, but the site must be secure before it can be deployed.

### **Some Basic Tests from the Ariba Buyer Application**

#### **Testing the PunchOut item**

Create a requisition in Ariba Buyer. Select a PunchOut item from the PunchOut catalog for the specific supplier.

*Expected behavior:* The system should connect to your site. The user should be able to shop, place items in a cart, then return the cart to the Ariba Buyer requisition.

#### **Simulating a lost connection**

Create a requisition in Ariba Buyer. Select a PunchOut item from the PunchOut catalog. The system brings up the supplier's site. Close the supplier's site before "checking out."

*Expected behavior:* The user should be returned to Ariba Buyer and see the Ariba Front Page.

Contact with the site reinitiates and a PunchOut item selected from the PunchOut catalog.

*Expected behavior:* Return to the supplier's site, where shopping cart should be empty.

#### **Testing multiple line items on requisition**

Create a requisition with two line items from the same shopping cart. Cart is returned to the requisition. Select one item to initiate the edit functionality.

*Expected behavior:* Both items appear in the cart upon return to the supplier's site.

The Cart returns to Ariba Buyer. Both items are selected and removed.

*Expected behavior:* All the items selected from the PunchOut session should be removed.

### **Testing a copied requisition with PunchOut items**

The Ariba Buyer copy function copies all the details on the line item, except for the SupplierPartAuxiliaryID. This preserves the integrity of the supplier's data, since you may use this element as a unique identifier for the quote. You must inform the buyer if they can fulfill orders submitted this way.

### **Testing a non-catalog purchase**

Since cXML is normally used for routing orders when PunchOut is implemented, the buyer and supplier must agree on how to handle and route non-catalog and ad hoc purchases.

### **Testing services basic functionality**

1. Using PunchOut, a user goes to the supplier site and provides configuration data. The supplier returns a line item.
2. Using PunchOut with an edit, the user selects a product which causes a line item to be brought back with pricing. Ariba CSN approval flow has been initiated. The order request is sent. The user now tries to PunchOut with both edit and inspect.

### **Testing contract pricing**

1. Using PunchOut, a buyer goes to your site and provides configuration data. You provide a line item back including contract pricing.
2. The Ariba user submits the requisition and initiates the workflow and approval process.
3. Once the requisition is approved, the user, through an edit PunchOut, selects a product. The user can also PunchOut with inspect. The supplier sends the order request.

### **Testing a configurable commodity**

1. Using PunchOut, the user goes to supplier site and selects configuration. Supplier provides line item(s) back to Ariba CSN.

2. The user can PunchOut with edit and inspect.
3. Ariba workflow and approvals are done through Ariba CSN. The order request is sent.

### **Test approver's ability to view and edit requisition**

1. Create a requisition in Ariba Buyer. Select a PunchOut item from the PunchOut catalog for the specific supplier.
2. Submit the requisition to initiate the workflow and approval process.
3. Log in as approver. PunchOut to view the PunchOut line item.
4. PunchOut to edit the PunchOut line item. Make a change to the line item.
5. Approve the requisition.

## **Deployment**

---

Prior to going live, have the buyer migrate content and order routing to production. On the supplier side, push the tested site to production. This includes creating new Websites, generation of customer-specific content and pricing, and infrastructure enhancements to support additional traffic. In addition, confirmation should be made that the Customer Service organization on your side is ready to support any new policies and procedures. Schedule several orders through the buyer's and supplier's system to validate connectivity in the production environment. Finally, have the buyer confirm your go-live date and closely monitor the Ariba account to ensure connectivity with the buyer's order entry system.

You should publish a standard operating procedure for bringing on new customers. A guideline for this document appears in Chapter 5. This document should contain information for the buyers so they are aware of how your integration works. Typically this will include topics such as:

- How you authenticate the user
- How you handle non-Catalog orders
- How you handle copied requisitions containing PunchOut date
- How to handle control data such as shipTo LocationID's from the buyer
- How the buyer can specify content to be shown, such as contract only, or both non-contract and contract items

The availability of this information will make the rollout to subsequent customers a much easier task.

---

# Chapter 2

## Expediting Development of a PunchOut Site

If speed to market is a concern, use Ariba's examples as a starting point. If ASP (Active Server Pages), Cold Fusion, or another scripted environment is already in place, then Website assistance is readily available. Ariba has some "non-supported" examples available on how to handle PunchOut Setup Request and Response documents. Once a system can support the Request and Response, an Ariba user can PunchOut to the supplier's Website. To obtain the example source code, request a zip file from an Ariba Commerce Services Network contact.

Use the following steps as a guide to help you to rapidly deploy a PunchOut site:

1. Remove all non-configuration related processes.
2. Adapt a bid/quote, then a receive order model.

If the current site won't support the order model, leverage the existing code base and build a new site with new processes. One size fits all is not always appropriate, and often looks and works poorly.

3. Remove or deactivate:

- Payment info
- Shipping info
- Workflow

4. Clean up the User Interface.

Remove all links to outside Websites. A PunchOut user should not be able to escape the PunchOut site through site navigation during a session.

## cXML Messaging

---

The PunchOut Setup Request and PunchOut Setup Response are how a buyer obtains the login to a supplier site and initiates a PunchOut session. See “Rapid Site Deployment” on page 52 of this document for sample cXML code geared toward rapid implementation.

A zip file of the code is available from Commerce Services for ASP (Active Server Page) and Cold Fusion implementations. See “Ariba Technical Support” on page vi for contact information.

**Note:** If using the ASP version, Microsoft IIS 3.0 or 4.0 and Internet Explorer 5 are required on the Web server.

The following is a clarification of the Active Server Pages referenced in the example.

- receivePunchoutSetupRequest.asp

This file grabs the HTTP POST cXML message and outputs a valid cXML Response document containing the supplier’s login URL.

- resolveXML.asp

This file loads the HTTP POST into a Microsoft DOM object and extracts data for the PunchOut Setup Response document.



---

# Chapter 3

## Retrofitting an Existing Website

In many cases, a supplier has an existing Website they may be able to use to accommodate PunchOut. This approach can be either very difficult or simple depending on what is currently in place. Software works best when its design meets its use. Requirements change, and opportunities evolve; the best of both worlds is to leverage what has already been created.

---

### Leveraging the Existing Site

---

Often suppliers have an existing e-commerce site that they consider a B2B site. Their customers typically log in directly, configure commodities, and place orders. The end user enters an order, usually pays by credit card, and selects shipping instructions. Then, the order is placed and the transaction is complete. However, technically, this is a B2C type application: consumer direct, commodity selection, and order creation.

If the site has dynamic workflow, approvals, saved shopping carts, and contracted pricing, then it's a B2B site. But even if the supplier has a true B2B type application, it still most likely images B2C, and there are several processes that are no longer needed. Existing processes such as dynamic workflow, or collection of shipping and payment information, will probably not be needed during configuration for an Ariba Buyer user.

The Ariba model to adopt when creating a PunchOut site is *bid/quote*, then *purchase*. PunchOut sessions are for configuration only. The process flow is as follows:

1. An Ariba Buyer user uses PunchOut to go to the supplier site and select a product.
2. The data is brought back to Ariba Buyer.
3. In Ariba Buyer, the user selects logistic information including *Bill-to*, *Ship-to*, *shipping method*, and *Need-by* date. This may also occur prior to the initial PunchOut session.

4. Approval process initiates. All required parties in the organization can inspect, edit, approve, or deny the requisition depending on each approver's role and their permissions.
5. The buying organization submits an Order Request document to the supplier through Ariba CSN. When the requisition is fully approved in Ariba Buyer, it is transferred to the supplier as cXML, EDI, e-mail, or fax. The Order Request is the source of truth for shipping, billing, and need-by date, along with other information contained on the Order Request cXML message. For further details, please reference the *cXML 1.1 User's Guide* at <http://www.cxml.org/home/>.

---

# Chapter 4

## Integrating Service Organizations

Often, tangible products are not the only items an Ariba Buyer user might wish to procure through PunchOut. Many companies supply only services. Developing a PunchOut site for services requires detailing the process with the customer.

Procuring services through Ariba Buyer is very different from procuring commodities such as books and pencils, which are more easily managed. For these simple commodities, the work flow for approval and the access control lists are within Ariba Buyer; there is no preliminary “configuration” required, and the number of items listed in the catalog are not in the tens of thousands, nor are they dynamically changing based on market conditions.

Any commodity that does not follow the above principles lends itself to PunchOut, where the commodity catalog is maintained by the supplier in the context of Ariba Buyer. Thus, integrating services with the Ariba Buyer requires the supplier to PunchOut enable their services exchange.

---

### PunchOut for Services

---

You can create PunchOut sites that supply services or contract work. The vast majority of supplier service exchanges operate on these two distinct workflow possibilities:

- Opening a position
- Engaging a candidate to fulfill the opened position

If the request to punch out is successful, the supplier’s exchange sends a URL to Ariba Buyer which opens a new browser window with that URL. All supplier session variables now pertain to the new browser window, where the Ariba Buyer user begins a new session, completely controlled by the supplier’s services exchange. After the user has initiated and completed the session within the services exchange, they are

sent back to Ariba Buyer. The supplier's services exchange submits the required fields to Ariba Buyer, populating a single line of the requisition. This first milestone in a PunchOut is called a "create" operation.

## Milestones

---

The two milestones involved in PunchOut are creating a session, and editing a session. Creating a session to retain a "contract worker" follows to further describe the process of these milestones:

- Create session

The PunchOut operation of opening a contract worker position is a *create* session. This session is subject to a fiscal approval flow within Ariba Buyer if it returns with dollar estimates for requested services.

The work flow can be stopped temporarily from proceeding by withholding a key field such as Unit of Measure (UOM). The Ariba user receives an e-mail notification from the supplier's services exchange to return to the site and continue with the process. At this point, asynchronous communications between the buyer and the supplier, initiated by the supplier, can exist only in the form of e-mail messages. This e-mail message from the supplier can only initiate an action by the Ariba user and cannot automatically trigger an event within Ariba Buyer.

- Edit session

The *edit* session is where the above mentioned missing UOM field can be entered, thus making the requisition complete and ready for the start of the approval process within Ariba Buyer. The *create* session is the first event and opens a contract worker position; *edit* is the second event and engages a candidate to fill the opened position. Opening a position does not go through an Ariba approval, while engaging a candidate does.

If opening a position requires a pre-approval in Ariba Buyer, then a custom double approval chain for the service-specific commodity needs to be written. The supplier's service exchange needs to be notified of approval or denial of this request. At this time, this can be done only through e-mail notifications, e-mail notifications cannot be triggered by the approval engine.

The *edit* operation is the more crucial step, since it populates Ariba Buyer with the required fields, while the *create* operation triggers the search for the candidate. After the approval process within Ariba Buyer, the Purchase Order is sent to the supplier's service exchange.

The contract worker starts on the negotiated start date and enters time worked into time sheets maintained and invoiced to Ariba Buyer, in this case the supplier of the service. The Accounts Payable ERP system linked to Ariba Buyer processes the invoices and pays against them accordingly. The AP process is outside of Ariba Buyer at this time and not tracked within Ariba Buyer.

Note that this is a simplified approach to a possible integration. This integration assumes a sign-off from all parties concerned. Any modification to this approach will require customizing and the involvement of an Ariba implementation team. Time sheets, for example, are not supported by standard PunchOut messages and require additional development by the supplier and the buyer.



---

# Chapter 5

## Creating a PunchOut Deployment Guide

Suppliers implementing PunchOut sites need to provide Procurement application and Marketplace implementors with a PunchOut Deployment Guide that explains their policies, capabilities, and processes. This chapter describes what should be included in this guide.

### Outline

---

The PunchOut Deployment Guide should contain the following sections:

- Connectivity Overview
  - PunchOut integration
  - Specific business rules/process flow
- Authentication and Identification
  - Buyer
  - Marketplace
- Required Extrinsic
  - User
  - CostCenter
  - Location
- Content Requirements/Specification
  - Categories
  - SelectedItem support
  - Contract items
  - Non-contract items
- Address Information
  - DeliverTo content
  - Address content

- Accounting Structure
  - Fields required for reconciliation
- Commodity Coding
  - Commodity code standards used
  - Granularity employed
- Transactions Supported
  - Change/Cancel Orders
  - Ad Hoc/NonCatalog Line Items within an Order
  - Internal Catalog Orders
  - Copied Requisitions
  - Order Exceptions
  - Third Party Suppliers
  - Quote Split into Multiple Orders

Each of these sections is described below.

## Connectivity Overview

---

In this section, describe the PunchOut process flow and the integration to Procurement and Marketplace applications. You can copy the explanation from the PunchOut Event Sequence section of the *cXML User's Guide*.

Include any application specific processes included in the integration. For example, document RFQ (Request For Quote) or Service Requisition functions where a second PunchOut may be required to receive the actual pricing of selected items.

## Authentication and Identification

---

In this section, explain how you perform authentication of PunchOut and how to use it. For Marketplaces, describe whether you authenticate both the Marketplace and the Marketplace participant.

Include any additional information used to identify the user, such as from the PunchOut Setup Request Extrinsic, or from the Contact or Address elements. Also describe the PunchOut site's ability to use this information to present custom content.



## Required Extrinsicics

---

If your site requires certain extrinsic information in order to initiate PunchOut, list those elements here. Keep the mandatory use of extrinsic elements to a minimum, because they increase implementation lead time.

For cXML 1.0 customers, User and CostCenter elements are often sent. However, because customers might call these something else, you should be prepared to accept this data under a different name.

For cXML 1.1 customers, user data is available from the Contact element, obsoleting the extrinsic elements User and CostCenter.

In Purchase Orders, extrinsicics are used to send additional information from the company at the line item level, such as Company Code and Contract number. Describe any line item extrinsicics required.

## Content Requirements/Specification

---

Describe the process and capabilities of content specification. Describe your process for selecting the categories of products that are shown to users. Also, if you have the capability to limit access to sections of the catalog to certain users within the organization, describe that here.

If you display both Contract and non-Contract items, describe how these are shown and who can see them. For instance, the typical user base might only be allowed to see the contract items in the catalog, while a purchasing agent is allowed to see the entire catalog. In this case, describe how your application determines the user's role.

Also, if you support the SelectedItem attribute (available only in cXML 1.1), describe how it works and provide the PunchOut Index File (either CIF or cXML) to the customer to have the appropriate links in their Ariba Buyer or Marketplace application.

## Address Information

---

The discussion of address format is important to make sure the procurement applications will interface with your order receipt application.

Some companies use the addressID attribute of the Address element to identify a pre-loaded ShipTo or BillTo in their application. If your company uses this, describe the process of loading and maintaining the customers' address data. Also, discuss how you handle exceptions, such as the user drop shipping the delivery to a location not already in their buyer application, or the user selecting a new location not known to the supplier. In the first case the addressID will be null, in the second case the addressID may be a number you do not have.

If you don't use the addressID attribute, determine what the customer is sending you in their address elements. This includes the DeliverTo elements, which are often the most problematic, since each customer chooses how to implement these. Most commonly, the use for these is a Person's name and/or their Building, Floor and Mailstop. Typically, two occurrences of the element are sent with the order, but some send only one. Collect the information the customer plans to send with these before implementation, so you will be prepared to map them accordingly into your system.

Similarly, the format and content of the Street elements need to be addressed. Determine from the customer the format of information the customer puts in these elements before implementation.

This will vary on a number of factors, including the ERP system used. A datasheet like the following is useful for capturing this information for discussion.

Element	#	Customer		Supplier		Ariba	
		Description	Max Length	Description	Max Length	Description (suggested)	Max Length
DeliverTo	1					Name	100
DeliverTo	2					MailStop/Bldg	100
Street	1					Combined length of 255 chars	
Street	2						
Street	3						
Street	4						

## Accounting Structure

Similar to the Address data, each customer has varying needs for the format of accounting information they send and receive for invoice reconciliation. This is due to a number of factors, including the ERP system used, and ultimately on the customer's GL design.

Collect this information during implementation and describe your ability to capture and return this information for invoicing and Pcard reconciliation.

A table like the following is useful for collecting the information that will come from your customers in their cXML message:

Segment	Customer Data (samples)			Return on Invoice
	Segment Type	Format / Length	Description	
Reporting Center	<i>Cost Center</i>	<i>Alpha 5</i>	<i>ID</i>	Y
Reporting Account	<i>Account</i>	<i>Integer 6</i>	<i>Account</i>	N
Reporting Business	<i>Company</i>	<i>Alpha 4</i>	<i>ID</i>	N

## Commodity Coding

Describe the commodity code standards you support for PunchOut and the level of granularity of the data that you return. Include an appendix that lists all the distinct commodity codes you will send, to which the customer can appropriately map before going live.

Also specify any specific commodity codes that you use to control workflow.

These commodity codes are defined by the UNSPSC (United Nations Standard Product and Service Code). For the list of UNSPSC codes, see [www.unspsc.org](http://www.unspsc.org).

## Transactions Supported

In this section, describe the electronic transactions that you support, the method you use to process orders, how you handle orders not supported electronically, and how you process exceptions. The following are specific order types you might need to describe:

## Change/Cancel Orders

Ariba CSN provides a separate method for routing change and cancel orders. You should describe how you route and process these messages. For example, a change/cancel order might be sent directly to shipping, which might hold the order, delaying processing.

## Ad Hoc/Non-Catalog Line Items within an Order

Non-catalog orders sent to you from Ariba Buyer or Marketplace come with a `SupplierPartID` of "Not Available". Describe if and how you intend to process these line items.

## Internal Catalog Orders

If you provide both static and PunchOut content to your customer, you might receive orders with line items from either or both of these formats. The customer sends them to the same order receiving method you define on Ariba CSN, so you need to be able to handle both. The difference between them is in the content of the `SupplierPartAuxiliaryID`. For PunchOut line items, you are sent the information you provided in the `SupplierPartAuxiliaryID` of the `PunchOutOrderMessage`. For the internal catalog orders, you are sent the content of the `SupplierPartAuxiliaryID` of the catalog item you sent them, or no data if you sent a catalog without this information.

## Copied Requisitions

Ariba Buyer allows customers to make copies of their requisitions for repeat orders. In the copy function, the contents of the entire line item is copied, except for the `SupplierPartAuxiliaryID`. This preserves the integrity of your system, because that element is in your control. Given that the `supplierPartID`, quantity, and price are all in the copied line item, you might still be able to process this line item. Describe in this section whether you will process the order.

## Order Exceptions

Because the price and/or SKU values can change between the time when the PunchOut session occurs and when the order is sent, describe the process for these exceptions.

## Third Party Suppliers

If your PunchOut site acts as a portal to other suppliers (an aggregator), and you do not receive those orders, list the suppliers and their D-U-N-S numbers for orders that should be sent to the supplier directly. To do this, you must supply the SupplierID element at the line item level. This ID must be registered on Ariba CSN and exist as a Common Supplier in the Ariba Buyer application. Prior to sending to Ariba CSN, Ariba Buyer determines to which supplier the OrderRequest is sent and populates the header with a To containing the D-U-N-S number of that supplier (extracted from the D-U-N-S number at item or requisition level).

## Quote Split into Multiple Orders

If your customer is paying with a PCard and assigns your products to various GL accounts, orders might arrive at your site containing only some items from a quote. While the entire quote is always approved or rejected as a unit, this split might cause exceptions in your order entry system. Describe in this section whether or not this situation will cause exceptions. If you cannot process the order, describe the process for these exceptions.

**Note:** The frequency of these situations will depend on the products displayed on your PunchOut site, how they are classified, and the buyer's accounting process.



---

# Chapter 6

## Sample Code

The examples provided in the following sections are meant only as a guide. The code below is not intended to be implemented “as is” into a supplier’s cXML documents or Website.

### cXML Index File

---

A cXML index file is a PunchOut index file that the buyer loads into Ariba Buyer to create entries in the catalog hierarchy. The file defines how the supplier and their products appear in the Ariba Buyer catalog UI. The important elements of an index file are as follows:

#### ***SupplierID***

Because buyers identify their respective suppliers in a unique fashion that is usually rooted in an ERP or legacy system (e.g., a vendor number), a supplier should choose a standard way of identifying themselves to buyers. The preferred standard among most buyers is the D-U-N-S® number. Information on D-U-N-S numbers can be found at <http://www.dnb.com/>.

#### ***URL***

The URL identifies a resource which processes a PunchOut Setup Request document. The Ariba Network, upon authenticating the setup request, may forward the cXML document to this resource. In all cases, this URL appears in the PunchOut Setup Request for sessions started in this line item.

#### ***Classification***

Buyers and suppliers often have unique ways of representing their items. For a punchout index item to map correctly to an entry in the buyer’s catalog hierarchy, items must be identifiable. The standard method among buyers for classifying products is UNSPSC. For more information on this, see <http://www.unspsc.com>.

Suppliers must identify the items they want represented in a buyer's catalog hierarchy by its UNSPSC code in this field. The number of items needed in the catalog hierarchy depends on the configuration of the buyer's system and on the supplier's ability to serve up a dynamic page based on product-level attributes. If the catalog is targeted for PunchOut only, only one default UNSPSC code is required. Note: in the index file the domain name is UNSPSC.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.009/cXML.dtd">

<Index>
  <SupplierID domain="DUNS">1234567</SupplierID>
  <Comments xml:lang="en-US">Sample cXML/Index</Comments>
  <IndexItem>
    <IndexItemPunchout>
      <ItemID>
        <SupplierPartID>po123456</SupplierPartID>
      </ItemID>
      <PunchoutDetail>
        <Description xml:lang="en-US">Supplier R Us</Description>
        <URL>http://www9.supp.com/cgi-
bin/hsrun/Staging/SRUS_SD/SRUS_SD.htx;start=AribaRequest</URL>
        <Classification domain="UNSPSC">88888889</Classification>
      </PunchoutDetail>
    </IndexItemPunchout>
  </IndexItem>
</Index>
```

## PunchOut Setup Request Document

---

A *PunchOut Setup Request* document authenticates ACSN and identifies the buyer for the supplier and passes the storefront URL supplied in the index file to the buyer.

### Key Elements

The important elements in a PunchOut Setup Request document are as follows:

#### ***Credential domain***

The <From> element contains the buyer's Network ID. The <To> element contains the supplier's D-U-N-S number.



**Identity**

Contains the Ariba Network ID and the supplier's D-U-N-S number.

**SharedSecret**

Contains the buyer's secret on the Ariba CSN which Ariba CSN replaces with the supplier's shared secret before forwarding.

**BuyerCookie**

Used by the supplier to track the actions of a buyer while in a particular PunchOut session. The cookie might change with *each* PunchOut session.

**Extrinsic**

Used to further identify a user to a supplier. The standard extrinsics sent from Ariba Buyer are User and CostCenter. Both elements are maintained by the buyer.

**Contact**

A person or group to contact for additional information.

**SupplierSetup URL**

Without a notification specified on [supplier.ariba.com](http://supplier.ariba.com), the SupplierSetup element specifies the URL of your PunchOut Website. This URL is generated from the Supplier's storeFrontURL, or from the PunchoutDetail URL, depending on where the user is when they PunchOut.

Otherwise, the e-commerce hub already knows the URL of your PunchOut Website.

In either case, this value, taken from the storeFrontURL or PunchoutDetail URL, appears in the document.

**SelectedItem**

Starting with cXML 1.1, procurement applications can use the SelectedItem element to specify store-, aisle-, or product-level PunchOut.

The SupplierSetup element has been deprecated. However, your PunchOut Website must continue to handle this method until all procurement applications generate the SelectedItem element.

## Sample

The following is a sample of a PunchOut Setup Request document:

```

<?xml version="1.0"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.009/cXML.dtd">
<cXML payloadID="958075346970@www.xyz3Demo.com " timestamp="2000-06-
14T12:57:09-07:00">
<Header>
  <From>
    <Credential domain="NetworkID">
      <Identity>AN01000002792-T</Identity>
    </Credential>
  </From>
  <To>
    <Credential domain="DUNS">
      <Identity>12345678-T</Identity>
    </Credential>
  </To>
  <Sender>
    <Credential domain="NetworkID">
      <Identity>AN01000002792-T</Identity>
      <SharedSecret>fH8 IcaIyx2</SharedSecret>
    </Credential>
    <UserAgent>Ariba ORMS 6.1</UserAgent>
  </Sender>
</Header>

<Request>
  <PunchOutSetupRequest operation="create">
    <BuyerCookie>1J3YVWU9QWMTB</BuyerCookie>
    <Extrinsic name="CostCenter">610</Extrinsic>
    <Extrinsic name="User">jpicard</Extrinsic>
    <BrowserFormPost>
      <URL>http:// buyeruser:3377/punchout</URL>
    </BrowserFormPost>
    <SupplierSetup>
      <URL>https://xyzdemo.xyzr.com/servlets/AribaBuyerServlet</URL>
    </SupplierSetup>
    <ShipTo>
      <Address addressID="001">
        <Name xml:lang="en">Acme Headquarters No2</Name>
        <PostalAddress>
          <DeliverTo>Jean Picard</DeliverTo>
          <Street>1565 Pine, MS A.2</Street>
          <City>New York</City>
          <State>NY</State>
          <PostalCode>01043</PostalCode>
          <Country isoCountryCode="US">United States</Country>
        </PostalAddress>
      </Address>
    </ShipTo>
  </PunchOutSetupRequest>
</Request>

```

```
</Address>
</ShipTo>
<Contact>
  <Name>jpicard</Name>
</Contact>
<SelectedItem>
  <ItemID>
    <SupplierPartID>54543</SupplierPartID>
  </ItemID>
</SelectedItem>
</PunchOutSetupRequest>
</Request>
</cXML>
```

## PunchOut Setup Response Document

---

After receiving a PunchOut Setup Request document, the supplier Website sends a *PunchOut Setup Response* document. The PunchOut Setup Response document serves two functions:

- Indicates whether the PunchOut Setup Request was successful
- Provides the procurement application with a redirect, or start page, URL to the supplier start page

### Key Elements

The important elements in a PunchOut Setup Response document are as follows:

#### **Status**

This element conveys the success or failure of a request operation. It is comprised of a code attribute and a text attribute, and an optional `xml:lang` attribute. The code attribute follows the HTTP status code model. In general, a 2xx series code indicates a successful client-server communication, a 4xx series code indicates a client error status code, and a 5xx series code indicates a server error code. The text attribute and optional `xml:lang` attribute allow for a text description of the status returned in a response. Suppliers are strongly encouraged to place the actual XML parse or application error in the body of the Status element. This allows for better one-sided debugging and interoperability testing.

## StartPage URL

The PunchOut Setup Response document contains a URL element that specifies the start page URL to pass to the user's browser for the interactive browser session.

## Sample

The following is a sample of a PunchOut Setup Response document:

```
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.009/cXML.dtd">
<cXML payloadID="958074700772@www.xyz3Demo.com" timestamp="2000-06-
14T12:59:09-07:00">
<Response>
  <Status code="200" text="success"/>
  <PunchOutSetupResponse>
    <StartPage>
      <URL>https://xyzdemo.xyzr.com/servlets/AribaServlet?pageid=psaribastartpage&#
x26;sessionId=986958074700612</URL>
    </StartPage>
  </PunchOutSetupResponse>
</Response>
</cXML>
```

## PunchOut Order Message Document

---

The *PunchOut Order Message* document provides product details and prices to the procurement application. You can also send hidden supplier cookies, which can later associate items with a specific shopping session.

Effectively, with the PunchOut Order Message, you have provided a quote for the requested items—but have not yet received a purchase order, so the order cannot yet be booked. This message is submitted to the procurement application, returning the user's browser to their requisition.

## Key Elements

The important elements in a PunchOut Order Message document are as follows:

***operationAllowed***

Specifies the PunchOut Setup Request operation allowed: create, inspect, or edit. This attribute controls whether the user can initiate a later PunchOut Setup Request transaction containing data from a PunchOut Order Message. If `operationAllowed=“create”`, only a later Order Request can contain these items. Otherwise, the procurement application can inspect or edit the shopping cart later, initiating subsequent PunchOut Setup Request transactions with the appropriate operations and the `ItemIn` elements corresponding to the `ItemIn` list returned in a PunchOut Order Message. Support for `edit` implies support for `inspect`.

***cxml-base64***

Hidden HTML form field that stores the cXML PunchOut Order Message. Optionally, this can be `cxml-urlencoded` in cXML 1.1.

***BuyerCookie***

Used by Ariba Buyer to validate the PunchOut Order Message and retrieve the line item collection. Once used, it is usually discarded by procurement application. The PunchOut Order Message *must* contain the same `BuyerCookie` that was used in the PunchOut Setup Request for this PunchOut session.

**Note:** The `BuyerCookie` might expire while the user is navigating your site. You might wish to support re-creation of a specific user’s last shopping cart. If provided, this must be an option and for only that user.

***SupplierPartAuxiliaryID***

Helps the remote Website transport complex configuration and bill-of-goods information to re-identify the item when presented to a remote Website in the future. If `SupplierPartAuxiliaryID` contains special characters, such as additional XML elements not defined in the cXML protocol, they must be escaped properly.

***Classification domain***

Specifies the commodity grouping of the line item to the buyer. All products and services must be mapped and standardized to the UNSPSC schema. Determines handling in ERP systems and buyer systems of different commodities with varying workflow. For a list of UNSPSC codes, see <http://www.unspsc.org/>

## Sample

The following is a sample of a PunchOut Order Message document:

```
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.009/cXML.dtd">
<cXML payloadID="958074737352@www.xyz3Demo.com"
timestamp="2000-06-14T12:59:09-07:00">
<Header>
  <From>
    <Credential domain="DUNS">
      <Identity>12345678</Identity>
    </Credential>
  </From>
  <To>
    <Credential domain="NetworkId">
      <Identity>AN0100002792-T</Identity>
    </Credential>
  </To>
  <Sender>
    <Credential domain="www.xyz3Demo.com">
      <Identity>PunchoutResponse</Identity>
    </Credential>
    <UserAgent>xyz3.x Buyer Services</UserAgent>
  </Sender>
</Header>
<Message>
  <PunchOutOrderMessage>
    <BuyerCookie>1J3YVWU9QWMTB</BuyerCookie>
    <PunchOutOrderMessageHeader operationAllowed="edit">
      <Total>
        <Money currency="USD">14.27</Money>
      </Total>
    </PunchOutOrderMessageHeader>
    <ItemIn quantity="2">
      <ItemID>
        <SupplierPartID>3171 04 20</SupplierPartID>
        <SupplierPartAuxiliaryID>ContractId=1751
ItemId=417714</SupplierPartAuxiliaryID>
      </ItemID>
      <ItemDetail>
        <UnitPrice>
          <Money currency="USD">1.22</Money>
        </UnitPrice>
        <Description xml:lang="en">ADAPTER; TUBE; 5/32"; MALE; #10-32 UNF;
FITTING</Description>
        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">21101510</Classification>
        <ManufacturerName>Dogwood</ManufacturerName>
      </ItemDetail>
    </ItemIn>
  </PunchOutOrderMessage>
</Message>
</cXML>
```

```

<ItemIn quantity="1">
  <ItemID>
    <SupplierPartID>3801 04 20</SupplierPartID>
    <SupplierPartAuxiliaryID>
      ContractId=1751 ItemId=417769
    </SupplierPartAuxiliaryID>
  </ItemID>
  <ItemDetail>
    <UnitPrice>
      <Money currency="USD">11.83</Money>
    </UnitPrice>
    <Description xml:lang="en">ADAPTER; TUBE; 5/32"; 2 PER PACK; MALE #10-
32 UNF; STAINLESS STEEL; FITTING</Description>
    <UnitOfMeasure>EA</UnitOfMeasure>
    <Classification domain="UNSPSC">21101510</Classification>
    <ManufacturerName>Legris</ManufacturerName>
  </ItemDetail>
  <SupplierID domain="DUNS">022878979</SupplierID>
</ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>

```

## Order Request Document

---

The *Order Request* is analogous to a purchase order. It contains header and line item details so that a product can be shipped and invoiced correctly to the buyer.

### Key Elements

In addition to the key elements identified in the Punchout Order Message, the following are critical in the Order Request document. In addition, this data might have changed since PunchOut.

#### ***ShipTo/DeliverTo***

##### **First line**

The person to whom product will ultimately be shipped.

##### **Second line**

The location, building, city, office, mail stop, where goods will be delivered.

**Note:** Not all buying organizations follow these recommendations for the use of the DeliverTo element.

## Sample

The following is a sample of a Order Request document:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.009/cXML.dtd">
<cXML version="1.0" payloadID="95807581568710.10.90.204" timestamp="2000-05-
11T13:10:15+07:00">
  <Header>
    <From>
      <Credential domain="NetworkId">
        <Identity>AN010000002792</Identity>
      </Credential>
    </From>
    <To>
      <Credential domain="DUNS">
        <Identity>12345678</Identity>
      </Credential>
    </To>
    <Sender>
      <Credential domain="NetworkId">
        <Identity>AN010000002792</Identity>
        <SharedSecret>fH8IcAIyx2</SharedSecret>
      </Credential>
      <UserAgent>Ariba Buyer 7.0 </UserAgent>
    </Sender>
  </Header>

  <Request>
    <OrderRequest>
      <OrderRequestHeader orderID="DO152" orderDate="2000-05-
11T13:10:11+07:00" type="new">
        <Total>
          <Money currency="USD">13.05</Money>
        </Total>
        <ShipTo>
          <Address isoCountryCode="US" addressID="001">
            <Name xml:lang="en">Acme&#032;Headquarters&#032;No2</Name>
            <PostalAddress name="default">
              <DeliverTo>Jonluc&#032;Picard</DeliverTo>
              <DeliverTo>Acme&#032;Headquarters&#032;No2</DeliverTo>
              <Street>1565&#032;Pine,&#032;MS&#032;A.2</Street>
              <City>New&#032;York</City>
              <State>NY</State>
              <PostalCode>01035</PostalCode>
              <Country isoCountryCode="US">United&#032;States</Country>
            </PostalAddress>
          </Address>
        </ShipTo>
      </OrderRequestHeader>
    </OrderRequest>
  </Request>
</cXML>
```



```

</PostalAddress>
<Email name="default">rlim@acme.com</Email>
<Phone name="work">
  <TelephoneNumber>
    <CountryCode isoCountryCode="US">1</CountryCode>
    <AreaOrCityCode>718</AreaOrCityCode>
    <Number>9306200</Number>
  </TelephoneNumber>
</Phone>
<Fax name="work">
  <TelephoneNumber>
    <CountryCode isoCountryCode="US">1</CountryCode>
    <AreaOrCityCode>718</AreaOrCityCode>
    <Number>9308410</Number>
  </TelephoneNumber>
</Fax>
<URL name="default">www.acme.com</URL>
</Address>
</ShipTo>
<BillTo>
  <Address isoCountryCode="US" addressID="15">
    <Name xml:lang="en">Acme&#032;Headquarters</Name>
    <PostalAddress name="default">
      <Street>1565&#032;Pine,&#032;MS&#032;A.2</Street>
      <City>New&#032;York</City>
      <State>NY</State>
      <PostalCode>01035</PostalCode>
      <Country isoCountryCode="US">United&#032;States</Country>
    </PostalAddress>
    <Email name="default">rlim@acme.com</Email>
    <Phone name="work">
      <TelephoneNumber>
        <CountryCode isoCountryCode="US">1</CountryCode>
        <AreaOrCityCode>718</AreaOrCityCode>
        <Number>9306200</Number>
      </TelephoneNumber>
    </Phone>
    <Fax name="work">
      <TelephoneNumber>
        <CountryCode isoCountryCode="US">1</CountryCode>
        <AreaOrCityCode>718</AreaOrCityCode>
        <Number>9308410</Number>
      </TelephoneNumber>
    </Fax>
    <URL name="default">www.acme.com</URL>
  </Address>
</BillTo>
</OrderRequestHeader>
<ItemOut quantity="1" >
  <ItemID>

```

```

        <SupplierPartID>3171&#032;04&#032;20</SupplierPartID>
        <SupplierPartAuxiliaryID>ContractId=1751&#032;ItemId=417714</SupplierP
artAuxiliaryID>
        </ItemID>
        <ItemDetail>
        <UnitPrice>
        <Money currency="USD">1.22</Money>
        </UnitPrice>
        <Description>
xml:lang="en">ADAPTER;&#032;TUBE;&#032;5/32&quot;;&#032;MALE;&#032;#10-
32&#032;UNF;&#032;FITTING</Description>
        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">88888889</Classification>
        <ManufacturerName>Legris</ManufacturerName>
        <URL>https://xyzdemo.xyzr.com/servlets/AribaBuyerServlet</URL>
        <Extrinsic name="PR&#032;No.">PR172</Extrinsic>
        <Extrinsic
name="Requester">Jonluc&#032;Lim&#032;(AN&#032;Consultant)</Extrinsic>
        </ItemDetail>
        <Distribution>
        <Accounting name="DistributionCharge">
        <Segment type="Cost&#032;Center" id="5202"
description="Department&#032;Name"/>
        <Segment type="Account" id="5203" description="Account&#032;Name"/>
        </Accounting>
        <Charge>
        <Money currency="USD">1.22</Money>
        </Charge>
        </Distribution>
        </ItemOut>
        <ItemOut quantity="1" >
        <ItemID>
        <SupplierPartID>3801&#032;04&#032;20</SupplierPartID>
        <SupplierPartAuxiliaryID>ContractId=1751&#032;ItemId=417769</SupplierPartAuxiliaryID
>
        </ItemID>
        <ItemDetail>
        <UnitPrice>
        <Money currency="USD">11.83</Money>
        </UnitPrice>
        <Description>
xml:lang="en">ADAPTER;&#032;TUBE;&#032;5/32&quot;;&#032;2&#032;PER&#032;PA
CK;&#032;MALE&#032;#10-
32&#032;UNF;&#032;STAINLESS&#032;STEEL;&#032;FITTING</Description>
        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">88888889</Classification>
        <ManufacturerName>Legris</ManufacturerName>
        <URL>https://xyzdemo.xyzr.com/servlets/AribaBuyerServlet</URL>
        <Extrinsic name="PR&#032;No.">PR172</Extrinsic>

```

```

    <Extrinsic
name="Requester">Jonluc&#032;Lim&#032;(AN&#032;Consultant)</Extrinsic>
</ItemDetail>
<Distribution>
  <Accounting name="DistributionCharge">
    <Segment type="Cost&#032;Center" id="5202"
description="Department&#032;Name"/>
    <Segment type="Account" id="5203" description="Account&#032;Name"/>
  </Accounting>
  <Charge>
    <Money currency="USD">11.83</Money>
  </Charge>
</Distribution>
</ItemOut>
</OrderRequest>
</Request>
</cXML>

```

## Order Response Document

---

The *Order Response* document acknowledges that you have received the purchase order and that it parsed correctly. It is not a commitment to execute the purchase order. If the message received is not a valid cXML document, a return code of 500 is acceptable. All cXML Order Request transactions, including ad hoc, changed, and cancelled orders, should be acknowledged with a Status code of 200.

### Sample

The following is a sample of a Order Response document:

```

<?xml encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://xml.cxm.org/schemas/cXML/1.1.009/cXML.dtd">
<cXML timestamp="2000-06-14T13:07:39-07:00" payloadID="958075659144--
4074645440380323973@206.251.25.163">

  <Response>
    <Status code="200" text="OK"></Status>
  </Response>
</cXML>

```

## Rapid Site Deployment

---

The following ASP examples can be used as a starting point to rapidly deploy a site for PunchOut. The two files demonstrate how to handle PunchOut Setup Request and Response documents. Once a system can support the Request and Response, an Ariba user can punch out to the supplier's Website. To obtain the example source code, request a zip file from an Ariba Commerce Services contact.

Please note that this is a basic example and not production quality. For example, credential variables attached in the query string are not acceptable for production code. Variables have not been declared and must be prior to use. Since a global variable file has not been included with this example, change the following variables in the appropriate locations:

### receivePunchoutSetupRequest.asp

```
myAribaPassword = "welcome"
myURL = http://nathanalbee/punchoutasp/
```

### resolveXML.asp

```
olddtdvalue = ""cXML.dtd""
newdtdvalue = ""http://nathanalbee/cxml1.1/cXML.dtd""
```

## File receivePunchoutSetupRequest.asp

This file grabs the HTTP POST cXML message and outputs a valid cXML PunchOut Setup Response document containing the supplier's login URL:

```
<%@LANGUAGE = VBScript%>
<%
    Dim myAribaPassword
    Dim myURL
%>
<%
*****Comments*****
'Include file that loads http post cXML and parses with MS DOM
*****
%>
<!--#include file="resolveXML.asp" --->
<%
*****Comments*****
' Variables needed for example Very simple validation for asp
' Set sharedsecret password expected from Ariba
*****
    myAribaPassword = "welcome"
```

```

        myURL = "http://nathanalbee/punchoutaspv2/"
    %>
<%
*****Comments*****
'Basic Shared Secret validation.
'IMPORTANT
'Please note: You will need to fix the timestamp and payloadID the
'f() Now is currently invalid and will fail on some builds of Ariba Buyer as well as
'other product lines. Please user format specified in the cxml users guide available
'@ http://www.cxml.org
'See file TCcXMLFormatDTime.asp for a good start on formatting the time ISO 8601
date/time standard.
*****

if (myAribaPassword <> sharedSecret) then %>
<?xml version="1.0" ?>
<!DOCTYPE cXML SYSTEM "cXML.dtd">
<cXML payloadID="&lt;%= Now &amp;&quot;@&quot;&amp;
Request.ServerVariables(&quot;LOCAL_ADDR&quot;)%&gt;" timestamp="&lt;%= Now
%&gt;">
<Response>
<Status code="500" Text="Invalid document" />
</Response>
</cXML>
<% else %>
<?xml version="1.0" ?>
<!DOCTYPE cXML SYSTEM "cXML.dtd">
<cXML payloadID="&lt;%= Now &amp;&quot;@&quot;&amp;
Request.ServerVariables(&quot;LOCAL_ADDR&quot;)%&gt;" timestamp="&lt;%= Now
%&gt;">
<Response>
<Status code="200" text="Success">
</Status>
<PunchOutSetupResponse>
<StartPage>
<URL>
<%=myURL%>

<p>b2bsite/shop.asp?fromIdentity=<%= fromIdentity%>&amp;operation=<%=
operation%>&amp;buyerCookie=<%= buyerCookie%>&amp;BrowserFormPost=<%=
BrowserFormPost%></URL></StartPage></PunchOutSetupResponse></Response></c
XML> <%end if%> </p>

```

## File resolveXML.asp

This file loads the HTTP POST into a Microsoft DOM object and extracts data for the PunchOut Setup Response document:

```

<%
Dim xml
Dim xdoc
Dim xml2
*****Comments*****
' MSDOM can't resolve local DTD's, so replace defination with URL
' GET DTD's from http://cxml.org/
' But put them locally on your server for performance
*****
Dim oldtdvalue
Dim newtdvalue
oldtdvalue = ""cXML.dtd""
newtdvalue = ""http://nathanabee/cxml1.0/cXML.dtd""

if (Request.ServerVariables("REQUEST_METHOD") = "POST") then
*****Comments*****
' This command reads the incoming HTTP cXML Request
' Note: this does not currently handle cXML documents sent
' with content-type: text/xml or any transfer which specifies
' the MIME charset attribute on that header.
*****
    totalBytes = Request.TotalBytes
    IF totalBytes > 0 THEN
        xml = Request.BinaryRead( totalBytes )
        for i = 1 to totalBytes
            xmlstr = xmlstr + String(1,AscB(MidB(xml, i, 1)))
        Next
        xml2 = xmlstr
        xml2 = Replace(xml2,oldtdvalue,newtdvalue)
        xml2 = Replace(xml2,"utf-8","utf-16")
        xml2 = Replace(xml2,"UTF-8","UTF-16")
    END IF

*****Comments*****
' Create MSDOM Object and set load values
*****

Set xdoc = Server.CreateObject("Microsoft.XMLDOM")
    xdoc.ValidateOnParse = True
    xdoc.async = False
    xdoc.resolveExternals = True
    loadStatus = xdoc.loadXML(xml2)

*****Comments*****
' Create MSDOM Object and set load values
' Note: thse XML retrievals handle only the first of any list of
' credential elements
*****

```

```

        If loadStatus = True then
            Set fromIdentity =
xdoc.getElementsByTagName("Header/From/Credential/Identity")
            fromIdentity = (fromIdentity.item(0).text)
            Set toSuppCred =
xdoc.getElementsByTagName("Header/To/Credential/Identity")
            toSuppCred = (toSuppCred.item(0).text)
            Set senderCred =
xdoc.getElementsByTagName("Header/Sender/Credential/Identity")
            senderCred = (senderCred.item(0).text)
            Set sharedSecret =
xdoc.getElementsByTagName("Header/Sender/Credential/SharedSecret")
            sharedSecret = (sharedSecret.item(0).text)
            Set fromUserAgent =
xdoc.getElementsByTagName("Header/Sender/UserAgent")
            fromUserAge = (fromUserAgent.item(0).text)
            Set operation =
xdoc.documentElement.childNodes(1).childNodes(0).attributes.getNamedItem("operation"
)
            operation =
xdoc.documentElement.childNodes(1).childNodes(0).attributes.getNamedItem("operation"
).text
            Set buyerCookie =
xdoc.getElementsByTagName("Request/PunchOutSetupRequest/BuyerCookie")
            buyerCookie = (buyerCookie.item(0).text)
            Set buyExtrinsics =
xdoc.getElementsByTagName("Request/PunchOutSetupRequest/Extrinsic")
            For i = 0 To (buyExtrinsics.length -1)
                BuyExtrinsicVars = (buyExtrinsics.item(i).text) & "," & BuyExtrinsicVars
            Next
            Set BrowserFormPost =
xdoc.getElementsByTagName("Request/PunchOutSetupRequest/BrowserFormPost")
            BrowserFormPost = (BrowserFormPost.item(0).text)

*****Comments*****
' Some nice MSDOM error logging for a failed parse or load
*****

        Else
            Response.Write " <P> xml @ supplier site failed to load using MSDOM: "
            Dim strErrText
            Dim xPE

            Set xPE = xdoc.parseError
            strErrText = "Your XML Document failed to load due the following error: " &
"Error #: " & xPE.errorCode & ":-" & "Line #: " & xPE.Line & "Line Position: " & xPE.linepos
& "Position In File: " & xPE.filepos & "Source Text: " & xPE.srcText & "Document URL: " &
xPE.url

            Response.Write strErrText
            End If

        Else

```

```

*****Comments*****
' ASP page was called using a GET. Functions expect a post.
*****
Response.Write "<P> Wrong Method Get: Post supported only"
End if
%>

```

## File TCcXMLFormatDTime.asp

This file contains a function to make the timestamp display correctly.

```

' Big thanks to John G. Lekas @ technicon.com for providing this function. ' This
function does not do the time adjustment but provides great start in helping you
out. ' <%
private function TCcXMLFormatDTime(strDTimeln)
'Purpose:
    'This function returns the inputted date/time value, formatted for
    'cXML messaging.
    'Currently, cXML uses the ISO 8601 date/time standard.
,
'Input:
    'strDTimelnInput date/time to be formatted.
,
'Ouptut:n/a
,
'Return:String value containing the formatted date/time.
,
    const conRoutine = "TCcXMLFormatDTime"

    dim strTempOutDtime
    dim strDay
    dim strMonth
    dim strTime

    TCcXMLFormatDTime = ""

    strTime = FormatDateTime(strDTimeln, 4)' Short time.

    ' Set the year.
    strTempOutDtime = Year(strDTimeln) & "-"
    ' Set the month.
    strMonth = Month(strDTimeln)
    if len(strMonth) = 1 then strMonth = "0" & strMonth
    strTempOutDtime = strTempOutDtime & strMonth & "-"
    ' Set the day and the date/time delimiter.
    strDay = Day(strDTimeln)
    if len(strDay) = 1 then strDay = "0" & strDay
    strTempOutDtime = strTempOutDtime & strDay & "T"
    ' Set the time.

```



```
strTempOutDtime = strTempOutDtime & strTime  
  
TcXMLFormatDTime = strTempOutDtime  
  
    exit function  
end function  
%> </p>  
</body>  
</html>
```



---

# Index

## A

- Ariba Ready 7
- Ariba Supplier Link 7
- ASL 7
- ASP 23

## B

- B2B 25
- B2C 25
- BuyerCookie element 41, 45

## C

- cancelled requisition 13
- Catalog Information Format 5
- CIF catalog v, 5, 8, 9, 17
- CIF files 1
- Classification element 39, 45
- Cold Fusion 23
- Contact element 41
- Credential element 11, 40
- Customer Advisory Council 7
- cXML 5, 26
- cXML catalog v, 9
- cXML index file 39
- cxml-base64 45
- cxml-urlencoded 45

## D

- Document Type Definition 5
- DOM object 24
- domain attribute 11, 40, 45
- DTD 5

- D-U-N-S number 4, 11, 18, 39

## E

- EDI 2, 26
- Extrinsic data 15
- Extrinsic element 41

## H

- HTTPS 19

## I

- Identity element 41

## N

- NetworkID 18

## O

- operationAllowed attribute 12, 45
- Order Request document 13
- Order Response document 13

## P

- Profile Request document 16
- PunchOut 1
- PunchOut catalog 1, 10, 17
- PunchOut Deployment Guide 31
- PunchOut index file 1, 39
- PunchOut Setup Request document 11
- PunchOut Setup Response document 12

PunchOutOrderMessage element 1

## R

receivePunchoutSetupRequest.asp 24, 52  
resolveXML.asp 24, 52

## S

SelectedItem element 14, 33, 41  
SharedSecret element 11, 41  
StartPage element 44  
Status element 13, 43  
SupplierID element 39  
SupplierPartAuxiliaryID element 15, 20, 45  
SupplierSetup element 41

## T

TCcXMLFormatDTime.asp 56  
Technical Support vi

## U

Unit of Measure 8, 28  
UNSPSC 5, 39  
UOM 8, 28  
URL element 39, 41, 44

## X

XML 5  
XML parser 5





**ARIBA, INC.**

1565 CHARLESTON ROAD  
MOUNTAIN VIEW, CA 94043 USA

TELEPHONE: 650.930.6200

FAX: 650.930.6300

[WWW.ARIBA.COM](http://WWW.ARIBA.COM)