Oracle® Procurement

Supplier's Guide to Punchout and Transparent Punchout

Release 11*i*

Part No. B13855-01

July 2004

ORACLE®

Oracle Procurement Supplier's Guide to Punchout and Transparent Punchout, Release 11*i*

Part No. B13855-01

Contents

2   Supplier Setup

3   Controlling Access to Punchout or Transparent Punchout

A Detailed Punchout and Transparent Punchout Process

B   DTDs, Documents, and Descriptions

C   Authentication, Security, and Encoding

D   Maximum Field Lengths

Index

Send Us Your Comments

Oracle Procurement Supplier's Guide to Punchout and Transparent Punchout, Release 11*i*

Part No. B13855-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appsdoc_us@oracle.com
- FAX: (650) 506-7200 Attn: Oracle Applications Documentation Manager
- Postal service:
  Oracle Corporation
  Oracle Applications Documentation Manager
  500 Oracle Parkway
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

x

Preface

Welcome to the Oracle Procurement Supplier's Guide to Punchout and Transparent Punchout, Release 11*i*.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.

- Your company's electronic catalog.

- XML or cXML, security issues, and character encoding.

See Other Information Sources for more information about Oracle Applications product information.

> **Note:** This guide covers Oracle Procurement Release 11*i*10 (Family Pack J) and Oracle Exchange Release 6.2.5.

How To Use This Guide

The Oracle Procurement Supplier's Guide to Punchout and Transparent Punchout describes how to use the punchout or transparent punchout capability of Oracle iProcurement and Oracle Exchange to provide access to remote catalogs.

This guide is intended for suppliers who wish to set up punchout or transparent punchout access to their catalogs for buyers who are using Oracle iProcurement or Oracle Exchange.

This guide contains the following chapters:

- Chapter 1 explains what punchout and transparent punchout are, describes the different models and the benefits of each, and helps buyers choose a model.

- Chapter 2 explains how a supplier implements a punchout or transparent punchout.

- Chapter 3 provides an overview of how both the buyer and supplier can control access to punchout and transparent punchout catalogs.

- Appendix A expands on the overview of each punchout and transparent punchout model by describing how the process works at a more detailed level.

- Appendix B displays the DTD used by punchout and transparent punchout, descriptions of the XML fields for the DTDs, and examples of punchout and transparent punchout documents.

- Appendix C provides a basic overview of the authentication and security in Oracle iProcurement and Oracle Exchange, and explains how these applications handle encoding in the punchout and transparent punchout XML documents.

- Appendix D provides the maximum field lengths that Oracle iProcurement and Oracle Exchange support for the transfer of data.

A separate guide, *Oracle Procurement Buyer's Guide to Punchout and Transparent Punchout*, exists for buyers.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Other Information Sources

You can choose from many sources of information, including documentation, training, and support services, to increase your knowledge and understanding of Oracle Procurement.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **PDF Documentation-** See the Online Documentation CD for current PDF documentation for your product with each release. This Documentation CD is also available on Oracle*MetaLink* and is updated frequently.

- **Online Help -** You can refer to Oracle Applications Help for current HTML online help for your product. Oracle provides patchable online help, which you can apply to your system for updated implementation and end user documentation. No system downtime is required to apply online help.

- **Release Content Document -** See the Release Content Document for descriptions of new features available by release. The Release Content Document is available on Oracle*MetaLink*.

- **About document -** Refer to the About document for information about your release, including feature updates, installation information, and new documentation or documentation patches that you can download. The About document is available on Oracle*MetaLink*.

Guides Related to This Product

Oracle Exchange Installation Guide

This guide describes how to install Oracle Exchange.

Oracle Exchange and Oracle Sourcing Company Administration Guide

This guide describes how administrators can set up Oracle Exchange or Oracle Sourcing for use by their company and its employees.

Oracle Procurement Buyer's Guide to Punchout and Transparent Punchout

This guide describes buyer-side setup for punchout and transparent punchout. It contains much of the same information in the Oracle Procurement Supplier's Guide to Punchout and Transparent Punchout that is equally relevant to buyers, but replaces the setup chapter with buyer-only setup steps.

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11*i*. It provides a useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

"About" Document

For information about implementation and user documentation, instructions for applying patches, new and changed setup steps, and descriptions of software updates, refer to the "About" document for your product. "About" documents are available on Oracle*MetaLink* for most products starting with Release 11.5.8.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff and describes the Oracle Application Object Library components that are needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. This manual also provides information to help you build your custom Oracle Forms Developer forms so that the forms integrate with Oracle Applications.

Other Implementation Documentation

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps licensed Oracle customers convert data from existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Oracle*Metalink*

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff master Oracle Procurement and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many education centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Procurement working for you. This team includes your technical representative, account manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle server, and your hardware and software environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

About Oracle

Oracle develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and applications products, along with related consulting, education, and support services, in over 145 countries around the world.

Your Feedback

Thank you for using Oracle Procurement and this user guide.

Oracle values your comments and feedback. In this guide is a reader's comment form that you can use to explain what you like or dislike about Oracle Procurement or this user guide. Mail your comments to the following address or call us directly at (650) 506-7000.

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Or, send electronic mail to appsdoc_us@oracle.com.

xx

1

Overview

This chapter covers the following topics:

- Benefits of Punchout and Transparent Punchout on page 1-1

- Choosing Between Punchout and Transparent Punchout on page 1-3

- Punchout on page 1-3

- Transparent Punchout on page 1-13

- Configurations and Re-Punchout on page 1-16

> **Note:** This guide covers Oracle iProcurement Release 11*i*10 (Family Pack J) and Oracle Exchange Release 6.2.5.

Benefits of Punchout and Transparent Punchout

The ability to access remote catalogs using punchout or transparent punchout benefits both the supplier and the buyer. It enables suppliers to maintain and host their own catalog information, while buyers can search for items from within their own Oracle iProcurement or Oracle Exchange system. The burden of maintaining the hosted catalog is removed from the buying organization, reducing catalog maintenance and data storage costs. In addition to the local catalog, both punchout and transparent punchout provide a single point of entry to catalog content regardless of where the content resides.

Remote catalogs are particularly useful for products that are configurable or include highly variable or dynamic items and pricing. These products are difficult and costly to maintain in a buyer-hosted (local) catalog. Catalogs with these kinds of items are better maintained by the supplier, to ensure the latest content and pricing

are available and to eliminate inefficiencies (such as purchase order revisions to correct pricing).

Remote catalogs may not suit every commodity type or supplier, however. The following table shows the kinds of catalog items remote catalogs are ideal for, as compared to local catalog items:

Table 1–1   Catalog Models

Model	Commodity (Catalog Items)
Local Catalog	Best suited for direct material (such as mass-produced mechanical parts); products with prenegotiated or stable prices; items for which blanket purchase agreements and quotations already exist in Oracle Purchasing; or indirect material that you want to manage locally.
Punchout or Transparent Punchout to Oracle Exchange	Best suited for indirect material (such as office supplies); standard maintenance, repair, and operation (MRO) items items; and products with stable pricing. (You can also link contract purchase agreements in Oracle Purchasing to items on Oracle Exchange.)
Punchout to Supplier	Best suited for products requiring a high degree of configuration (such as computer hardware or office furniture) and specialized services (such as printing or media services). For example, the supplier site may have special features unique to the industry that the punchout can use. (In an XML punchout, the supplier can also link its items to contract purchase agreements in Oracle Purchasing.)
Transparent Punchout to Supplier	Best suited for products with fluctuating prices, or extremely large or specialized catalogs, such as chemical supplies, that you want the supplier to manage. (The supplier can also link its items to contract purchase agreements in Oracle Purchasing.)

> **Note:**   A local catalog allows all searching methods: standard, expanded, and advanced searching; filtering and sorting search results; and browsing categories. A transparent punchout catalog allows standard searching and sorting by price. (A punchout catalog uses whatever search capabilities and tools the external site provides.)

Punchout supports both cXML and Oracle native XML standards, depending on the punchout model used. Transparent punchout supports Oracle native XML only. (The model names indicate whether the model supports XML, cXML, or both.)

Oracle native XML is XML that Oracle has adapted to today's business needs, as shown by the document type definitions (DTDs) and XML samples in this guide. For more information on cXML, see http://www.cxml.org.

Choosing Between Punchout and Transparent Punchout

Punchout and transparent punchout offer the same basic benefits—mainly, the benefit of the supplier maintaining its own item information, reducing maintenance costs for the buyer and supplier, and a single point of access to catalog content regardless of where it resides.

The main distinctions between punchout and transparent punchout are as follows:

- With transparent punchout, the requester does not visibly access the supplier site. Transparent punchout accesses the site in the background and returns the items directly to Oracle iProcurement, with no changes to the requester's user interface. Requesters perform no additional navigation to return to Oracle iProcurement.

- Whereas transparent punchout is ideally suited for products with fluctuating prices, punchout is ideally suited for configurable products. In addition, the supplier's site may have special features unique to the industry or to the buyer-supplier relationship that a punchout can take advantage of.

In a transparent punchout, the supplier sets up integration with its search engine to properly return search results to Oracle iProcurement. In a punchout, the supplier sets up integration to access its catalog and provide a mechanism for returning the requester to Oracle iProcurement. (If the supplier has already implemented a punchout, the supplier can leverage some of that setup when implementing a transparent punchout.)

Punchout

Punchout enables buyers to click a link that goes to a supplier's catalog, search for items on the supplier's site, and return those items directly to the buyer's shopping cart. If punching out from Oracle Exchange, buyers add the supplier's items to their shopping cart on Oracle Exchange. If punching out from Oracle iProcurement, requesters add the external items to their shopping cart in Oracle iProcurement.

Punchout offers seven models:

- Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)

- Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML)

- Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML)

- Model 3a: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML)

- Model 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (cXML)

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

Punchout Models and Comparisons

Oracle iProcurement and Oracle Exchange support various punchout models. It is important for buyers and suppliers to decide on the model they want to use before implementing the punchout.

The following table compares the punchout models:

Table 1–2   Punchout Model Benefits

Model	Buyer Benefits	Supplier Benefits
Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)	iProcurement requesters have access to all supplier catalogs defined on Exchange. They have a similar user interface experience as in iProcurement.	Suppliers load their catalog items once on Exchange and reach many iProcurement customers in addition to the buyers already registered on Exchange. The Exchange catalog supports features such as buyer or group pricing (the ability to create prices visible only to a specified buyer or group of buyers), pricing approval by the buyer, price breaks, price effective dates, and automated catalog loading by the supplier.
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	This model offers a unique, point-to-point solution between the buyer and supplier.	Supplier closely manages the content. Supplier can control access by allowing only certain buyers to access the site. (Any punchout changes such as the catalog URL's changing, however, must be communicated to all buyer organizations.) Suppliers already maintaining cXML catalogs can continue to use those.
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Buyers have access to a centralized collection of punchouts simply by registering with Exchange. Purchasing Managers on Exchange can control buyer access to specific punchouts using the **Control Punchout Access** page.	Suppliers only have to define their punchouts on Exchange once and reach multiple customers. Suppliers can control the visibility of their punchout definitions by publishing or unpublishing the punchout on Exchange. Instead of or in addition to creating a punchout link on Exchange, the supplier can also load catalog items directly to Exchange and take advantage of the special pricing features of Exchange. (See the benefits for Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML) above.) Suppliers already maintaining cXML catalogs can continue to use those.
Models 4 and 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML or cXML)	iProcurement requesters have access to a centralized collection of punchouts. iProcurement administrator does not have to configure a punchout for each supplier, but can just download the supplier's punchout definition from Exchange.	Suppliers only have to define their punchouts on Exchange once, rather than configuring punchout separately for each iProcurement customer. Suppliers can control the visibility of their punchout definitions by publishing or unpublishing the punchout on Exchange. Suppliers already maintaining cXML catalogs can continue to use those.

An additional punchout option is the *double punchout*. A double punchout goes from Oracle iProcurement to Oracle Exchange, then from Oracle Exchange to the supplier. In a double punchout, the requester clicks a punchout link to Oracle Exchange and shops on Oracle Exchange; the Exchange itself contains punchout links to suppliers directly, and the requester can additionally click any of these. The details and setup for the double punchout are the same as those for Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML), plus Model 3a or 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML). The double punchout is sometimes used as a diagnostic tool for the via-Exchange punchouts, to see whether a particular problem is happening during the punchout *to* Oracle Exchange or *from* Oracle Exchange. The only difference between a double punchout and a via-Exchange punchout is that in a via-Exchange punchout, the via-Exchange step is invisible to the requester; in a double punchout, the punchout to Oracle Exchange is one (visible) step, and punching out from Oracle Exchange to a supplier's catalog is a second visible step.

Choosing Your Punchout

If you need more help choosing a punchout model, use the following diagram:

Figure 1–1   Punchout Model Decision Tree
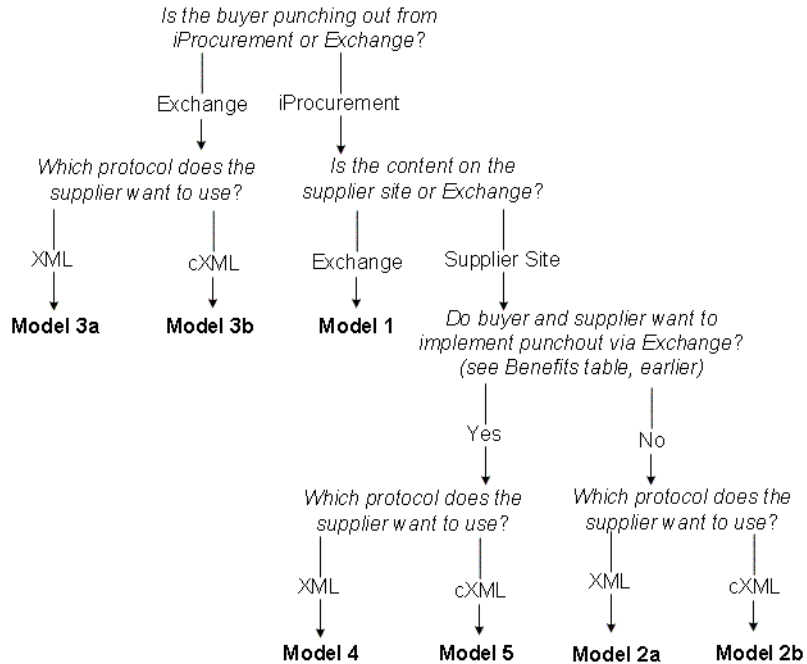
After you have selected your model, see Chapter 2 for a setup checklist specific to that model.

Punchout Flow

The following illustration shows a punchout from Oracle iProcurement on the **Shop** home page:

The following illustration shows a punchout from Oracle iProcurement on the **Search Results Summary** page, when a punchout matches the search keywords entered for a store:

ORACLE iProcurement

Shop | Requisitions | Receiving | Contractors

Stores | Categories | Shopping Lists | Non-Catalog Request | Contractor Request

Search Office Supplies [   ] (Go) Advanced Search   Shop Other Stores

Results from local or transparent punchout catalog

Search Results Summary from Regular Office Supplies: folder ←——————

View all results from Regular Office Supplies

(Hide Images)

Standard Classification Folders, Legal, Blue

Ideal for case histories, tax records, sales records, etc. Sturdy, 25-Point covers are made of a heavyweight durable Pressboard bonded with long-lasting Tyvek® gussets. 2"" metal fasteners are on the 2, 17 pt. kraft inner partitions.

Category:	**File Folders**	Supplier:	**Acme Supplies**	Supplier Item:	**FDR-0008**
Manufacturer:	**National Supplies**	Manufacturer Item:		Contract Number:	
Price:	**4.95 USD**	Unit:	**Each**		

Quantity [1] (Add to Cart)   (Add to Favorites) (Add to Compare)

Hanging Partition Fastener Folders, Ruby Red

Durable 25-point pressboard covers in 4 bright colors. 6 separate filing sections for documents and printouts. Sturdy kraft dividers with strong metal fasteners. Tear-resistant Tyvek gussets allow for 2 1/4"" expansion. Adjustable tab for easy identification.

Category:	**File Folders**	Supplier:	**Acme Supplies**	Supplier Item:	**FDR-0007**
Manufacturer:	**National Supplies**	Manufacturer Item:		Contract Number:	
Price:	**5.99 USD**	Unit:	**Each**		

Quantity [1] (Add to Cart)   (Add to Favorites) (Add to Compare)

Note: In an actual Search Results Summary page, the first three matching items would be displayed.

Search Results Summary from Oracle Exchange: folder ←————

Results from Oracle Exchange punchout catalog

(Hide Images)

Oracle Exchange

Visit exchange.oracle.com to find all of your office supplies.

Exchange

Search Results Summary from Acme Office: folder ←————

Results from supplier site punchout catalog

(Hide Images)

Acme Office

Visit the Web site of Acme Office to find office supplies and furniture. Acme Office also offers color copy services.

Shopping Cart
Your cart is empty.

Compare Items
No items selected.

Catalog Language
Your current catalog language:
American English
Change Catalog Language

In Oracle Exchange, the punchout links to supplier sites display in the Shop Supplier Sites box on the **Shopping** home page. These links also appear in other areas of Oracle Exchange.

The following illustration shows a punchout from Oracle Exchange:

The following illustration gives a basic overview of the punchout process, using a punchout from Oracle iProcurement to the supplier:

Figure 1–2   Example Punchout Flow from Oracle iProcurement to Supplier
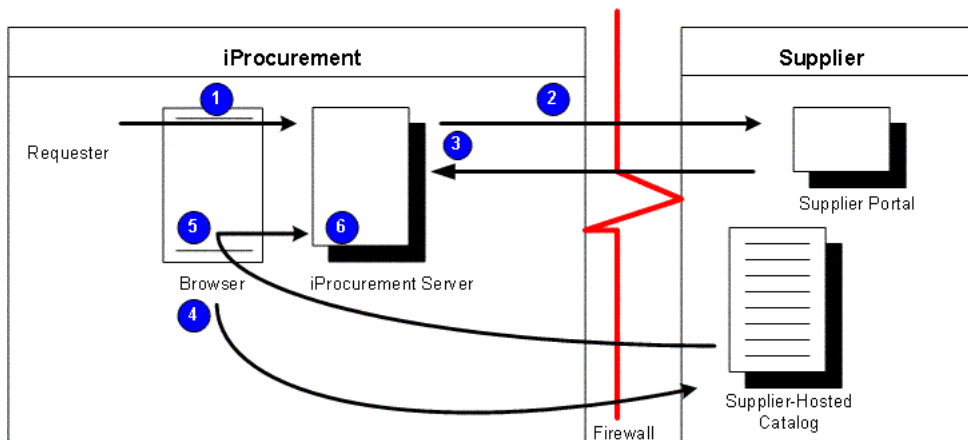
For an illustration of a punchout using Oracle Exchange, see Appendix A.

At certain steps in the punchout process, the details differ by model, but the basic flow is the same:

1. The requester in Oracle iProcurement or buyer in Oracle Exchange clicks a punchout link to the external catalog site.

   In Oracle iProcurement, punchout links are available from the following pages:

   - **Shop** home page when the requester clicks a store whose only catalog is a punchout.

   - **Shop Store** page when the requester clicks a store that contains a punchout catalog.

   - **Search Results Summary** page when a punchout matches the search keywords.

2. The *punchout from* application (Oracle iProcurement or Oracle Exchange) sends the login request to the catalog site.

3. The *punchout to* application (Oracle Exchange or the supplier site) authenticates the requester or buyer and returns a login response.

4. The *punchout from* application redirects the requester's or buyer's browser to the catalog site.

5. The requester or buyer browses or searches for items on the external catalog site and completes shopping on the site.

6. Via the requester's or buyer's browser, the *punchout to* application returns the shopping cart with the items to the *punchout from* application.

   The requester or buyer completes the checkout process for the items in the shopping cart, and the *punchout from* application processes the order.

Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)

In this model, the supplier loads its catalog items directly to the Oracle Exchange catalog, the Oracle iProcurement administrator sets up Oracle iProcurement to use Oracle Exchange as the punchout hub, and requesters see a punchout link to Oracle Exchange displayed in their search results.

After clicking the punchout link to Oracle Exchange, the requester sees the Oracle Exchange **Shopping** home page.

For detailed descriptions of this model's flow, see Appendix A.

Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)

In this model, the supplier hosts the catalog at its own site or Web store, the Oracle iProcurement administrator sets up Oracle iProcurement to use the supplier as a punchout site, and requesters see a punchout link to the supplier site displayed on the Oracle iProcurement home page.

For detailed descriptions of this model's flow, see Appendix A.

Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)

In this model, the supplier hosts the catalog at its own site or Web store, and buyers registered on Oracle Exchange see a punchout link to the supplier site from the **Shopping** home page or **Search Results** page on Oracle Exchange. In addition to providing punchouts to Oracle Exchange registrants, suppliers often use this model to test a punchout to their site before buyers implement any of the other models.

For detailed descriptions of this model's flow, see Appendix A.

Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

In this model, the supplier hosts the catalog at its own site or Web store, which the requester in Oracle iProcurement accesses (invisibly) via Oracle Exchange. Using Oracle Exchange for the punchout simplifies the initial setup process and the authentication and maintenance of the punchout. The supplier must set up a punchout from Oracle Exchange to its Web store. The Oracle iProcurement administrator, using the eContent Manager in Oracle iProcurement, must download the supplier's punchout definition from Oracle Exchange.

For detailed descriptions of this model's flow, see Appendix A.

Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

In this model, the supplier hosts a cXML catalog at its own site or Web store, which the requester in Oracle iProcurement accesses (invisibly) via Oracle Exchange. Using Oracle Exchange for the punchout simplifies the initial setup process and the authentication and maintenance of the punchout. The supplier must set up a punchout from Oracle Exchange to its Web store. The Oracle iProcurement administrator, using the eContent Manager in Oracle iProcurement, must download the supplier's punchout definition from Oracle Exchange.

For detailed descriptions of this model's flow, see Appendix A.

Transparent Punchout

A transparent punchout (also known as a distributed search) allows requesters to search for items on an external site without leaving Oracle iProcurement. Unlike punchout, requesters do not access the site directly. Instead, when the requester searches for items, the transparent punchout works in the background to access the external site and return the matching items directly to the **Search Results** page. Requesters do not necessarily know the items came from an external site. From the **Search Results** page, requesters add the items returned from the transparent punchout to their shopping cart just as they would any other item in the local catalog.

Transparent punchout offers two models:

- Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)

- Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)

Transparent Punchout Models and Comparisons

Oracle iProcurement supports two transparent punchout models. The following table compares the models:

Table 1–3   Transparent Punchout Model Benefits

Model	Buyer Benefits	Supplier Benefits
Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)	Requesters have access to all supplier catalog items defined on Oracle Exchange.	Suppliers load their catalog items once on Oracle Exchange and reach many Oracle iProcurement customers in addition to the buyers already registered on Exchange.
		The Exchange catalog supports features such as buyer or group pricing (the ability to create prices visible only to a specified buyer or group of buyers), pricing approval by the buyer, price effective dates, and automated catalog loading by the supplier.
		Note: Transparent punchout does not honor price breaks that the supplier created on Oracle Exchange.
Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)	This model offers a unique, point-to-point solution between the buyer and supplier.	Supplier closely manages the content. Supplier can control access by allowing only certain buyers to access the site. (Any changes such as the site's URL's changing, however, must be communicated to all buyer organizations.)

Choosing Your Transparent Punchout

Whether to conduct a transparent punchout to Oracle Exchange or to the supplier depends on whether the supplier's catalog content exists on Oracle Exchange or its own site. For example, for more dynamic pricing and item information, the supplier may choose to host the content on its own site.

If the supplier chooses to host its content on Oracle Exchange, the supplier does not have to implement its own transparent punchout; Oracle Exchange has the transparent punchout capability built in. Oracle Exchange also provides buyer and group pricing, pricing approval by the buyer, and price effective dates. If the supplier's site does not offer these features, the buyer and supplier can decide to use a transparent punchout to Oracle Exchange.

Transparent Punchout Flow

The following diagram shows the basic flow of transparent punchout:

Figure 1–3   Models 1 and 2: Transparent Punchout to Oracle Exchange or Supplier (XML)
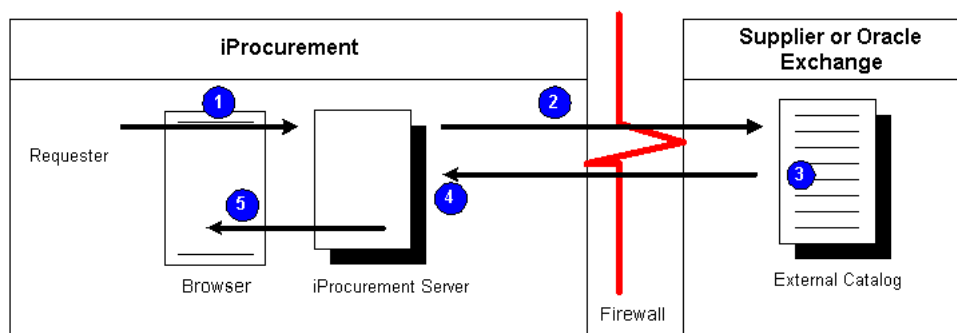
If the purchasing administrator has configured stores in the eContent Manager, the requester selects a store before searching.

The basic flow is as follows:

1. The requester in Oracle iProcurement conducts a search.

2. If the store includes a transparent punchout catalog, Oracle iProcurement sends a search request XML document to the external site with which the transparent punchout has been set up.

3. The external site authenticates the request, processes the request, and generates search results.

4. The external site returns a search results XML document to Oracle iProcurement.

5. Oracle iProcurement parses the search results XML document from the external site to display search results, as shown in the following illustration.

After viewing the transparent punchout results, the requester clicks Add to Cart for the desired items. The requester then completes the checkout process for the items in the shopping cart, and Oracle iProcurement processes the order.

For detailed descriptions of each step, see Appendix A.

Configurations and Re-Punchout

If the supplier site uses configuration numbers, then Oracle iProcurement and Oracle Purchasing carry that number onto the requisition and purchase order. In

addition, if the supplier allows re-punchout, requesters or approvers in Oracle iProcurement can view the configuration details later, using a re-punchout.

Configurations

In the punchout shopping cart, the supplier can specify a configuration number, in addition to the item number. For example, personal computers are often configurable and come with a configuration number. Oracle iProcurement assumes that configuration numbers are unique for each configuration, rather than a predefined configuration agreed upon by the buyer and supplier. (A predefined configuration should be treated as a standard item number.)

The <supplierReferenceNumber> field in the XML shopping cart and the <SupplierPartAuxiliaryID> field in a cXML shopping cart allow the supplier to identify the exact configuration that the requester selected. If the supplier specifies a configuration number, then this number travels with the requisition in Oracle iProcurement and with the purchase order in Oracle Purchasing. In both applications, the number is a hidden field called Supplier Config ID. (The number is also included in Oracle iSupplier Portal as a hidden field.)

Once the Supplier Config ID is returned to Oracle iProcurement, it cannot be edited, by neither buyer nor supplier, in neither Oracle iProcurement nor Oracle Purchasing. These applications do not check whether the Supplier Config ID is still valid during purchase order creation, but assume that it is. Only standard purchase orders support the Supplier Config ID field.

The configuration number is supported by the following models only:

- Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML)

- Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML)

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

Figure 1–4   Configuration Flow
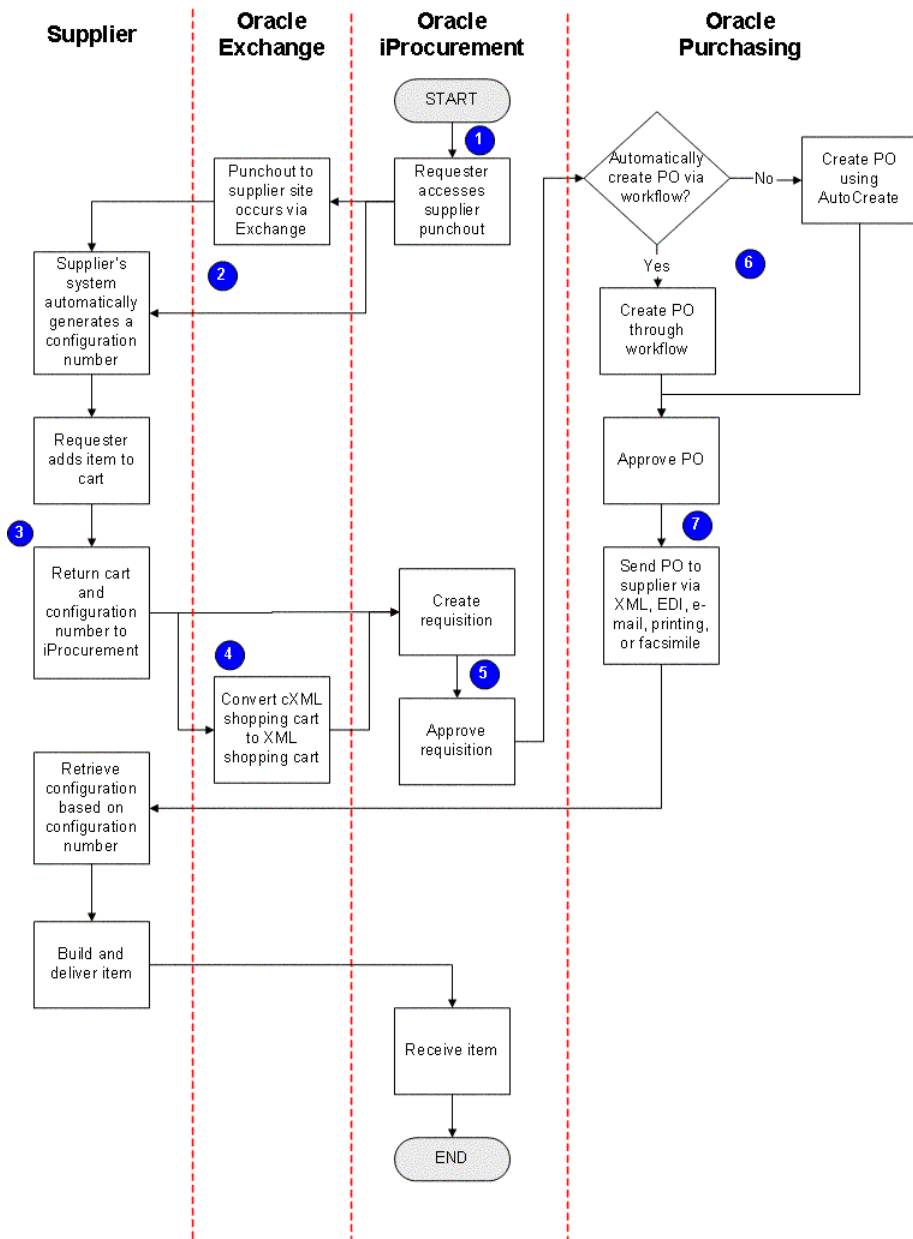
Figure 1–4 shows the following process for configuration numbers:

1. The requester accesses the punchout in Oracle iProcurement.

2. For the models that use Oracle Exchange, the punchout goes through Oracle Exchange. Otherwise, the punchout goes directly to the supplier, using XML or cXML.

3. The supplier site populates the shopping cart with the configuration number in the <SupplierPartAuxiliaryID> cXML field or the <supplierReferenceNumber> XML field.

   For examples, see the XML shoppingCart on page B-23 or the cXML PunchOutOrderMessage on page B-40.

4. For Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML), Oracle Exchange converts the cXML cart to an XML cart, transferring the <supplierPartAuxiliaryID> cXML value to the <supplierReferenceNumber> XML value.

5. The requester completes the checkout process, after which the requisition is created and approved.

6. Oracle Purchasing creates the purchase order (PO) automatically using Oracle Workflow, if the buyer's implementation uses workflow to automatically create purchase orders. Otherwise, the buyer uses the AutoCreate window in Oracle Purchasing to place the requisitioned item on a purchase order. The Supplier Config ID is a hidden field in the AutoCreate window and in the Purchase Order Lines Summary window.

7. In Oracle Purchasing, the purchase order is approved and sent to the supplier. The configuration number is included in XML, electronic data interchange (EDI), e-mail, print, and facsimile versions of the purchase order. (It is not "hidden" in these formats.)

The following illustrations show the configuration number (Supplier Config ID) hidden field in Oracle iProcurement:

In Oracle iProcurement, the administrator can use the personalization feature to add the Supplier Config ID column to the **Shopping Cart** page, if the procurement department wants requesters to see this information. The administrator can also personalize the Supplier Config ID to display when viewing or editing the requisition, approving the requisition, and receiving the requisition, and when viewing the purchase order in Oracle iSupplier Portal.

> **Note:** A personalization link displays only if the Oracle Applications Framework personalization profile option is set to display it. See the *Oracle Applications System Administrator's Guide* for more information.

Re-punchout

A re-punchout enables the requester or the requester's approver to revisit the configuration details of an item, after the initial punchout. For example, the requester can save the requisition and come back to it later. If re-punchout is allowed, then the item description is hyperlinked so that the user can click it, punch out to the supplier site (the punchout occurs in a new browser window), and view (not change) the configuration details of the item.

> **Note:** Re-punchout is supported by Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML) only.

The following illustration shows how the Item description is hyperlinked when a re-punchout is allowed:

Re-punchout must be set up by the supplier. If the `operationAllowed` header attribute in the shopping cart cXML document that the supplier returns is set to "inspect," then a re-punchout is allowed (the item description is hyperlinked). If it is set to "create," then a re-punchout is not allowed. The supplier can also specify "edit," but Oracle iProcurement treats this like the "inspect" option. Oracle iProcurement does not support editing configuration details, only viewing them.

If enabled, re-punchout is available in Oracle iProcurement when viewing or adding to the shopping cart; when creating, viewing, changing, or approving the requisition; and when receiving against the requisition.

The re-punchout process flows as follows:

1. A requester accesses a supplier site via punchout (Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML) only).

2. The requester selects an item and returns to Oracle iProcurement.

   In the cXML shopping cart, PunchOutOrderMessage, the supplier returns a value of "inspect" or "edit" in the `operationAllowed` attribute (understanding that "edit" is treated the same as "inspect").

3. Later, the requester or approver wants to view the configuration details of the punchout item.

   Since re-punchout is enabled, the item description is hyperlinked, and the requester or approver can click the link.

4. A new browser window opens, which accesses the supplier site using re-punchout.

   During the re-punchout, Oracle iProcurement sends a cXML request, PunchOutSetupRequest. The request includes the following:

   - The same user login information that is always contained in a PunchOutSetupRequest from this requester.

   - An `operation` attribute with the value that the supplier specified (either "inspect" or "edit").

   - The supplier item number in the <SupplierPartID> field.

   - The configuration number, if one exists, in the <SupplierPartAuxiliaryID> field. (There does not have to be a configuration number.)

5. The requester or approver can view the configuration details.

   Based on the login and item information, the supplier site displays the configuration details of the selected item.

   > **Note:** Even if the supplier allows the requester to change the configuration of the item on the supplier site, Oracle iProcurement does not return those changes. If the requester wants to make changes, the requester should delete the configured item from the shopping cart and access the punchout again via the **Shop** page.

6. When finished viewing the configuration details, the requester or approver closes the browser window and continues his or her task in Oracle iProcurement.

Suppliers should refer to the *cXML User's Guide* available at http://www.cxml.org/ for more details on the `operationAllowed` and `operation` attributes.

Re-punchout cannot be performed from within Oracle Purchasing, only in Oracle iProcurement.

2

Supplier Setup

This chapter covers the following topics:

-
-
-

The supplier must perform the steps in this chapter to ready its catalog for punchout or transparent punchout. The steps in this chapter apply to both XML and cXML suppliers.

> **Note:** Buyer setup is described in a separate guide solely for buyers, *Oracle Procurement Buyer's Guide to Punchout and Transparent Punchout*.

Setting up a punchout or transparent punchout requires some expertise in XML or cXML (if the supplier is using cXML), security (deciding whether to use SSL, for example), and character encoding. How long it takes the supplier to implement the punchout or transparent punchout depends in part on the supplier's expertise in these areas.

Setting Up Punchout

This section describes everything the supplier needs to do to set up any punchout model.

Punchout Implementation Considerations

It is important to decide on the model that you and your buyer want to use before implementing the punchout. Which model you use depends on where your catalog content resides (your own site or on Oracle Exchange) and whether you use XML or cXML. See Chapter 1.

- If you are implementing Model 3a or 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML), and your currency codes differ from the Oracle Exchange codes, you must perform data mapping on Oracle Exchange. See Create Data Mappings on Oracle Exchange on page 2-12.

- Decide how to validate user access of your site. Look ahead at Process Incoming XML or cXML Request on page 2-15. See also Chapter 3.

- It is strongly recommended that you use UTF-8 encoding when sending punchout documents. See Encoding on page C-4 for more information.

- Oracle iProcurement has requirements for item information. Read Appendix B to understand item requirements as described by the document type definitions (DTDs). For example, all items in Oracle iProcurement must have a numeric price; if not, the requester cannot add the item to the cart.

- The punchout setup typically creates one logon user account that is used by all requesters in the company to access the supplier site. Therefore, it is recommended that the supplier develop a method that allows multiple requesters with the same login account to access its site at one time. The supplier can accomplish this by making its site *session-aware*. Details are provided in Provide XML or cXML Response on page 2-16.

- Decide whether you will return configuration numbers (in addition to item numbers) to Oracle iProcurement. If so, decide whether you will allow re-punchout if you are implementing Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML). See Configurations and Re-Punchout on page 1-16.

- Decide how to redirect users to the *punchout from* application (Oracle Exchange or Oracle iProcurement). Look ahead at Determine Method for Returning Buyers to Procurement System on page 2-18.

Suppliers of licensed Oracle iProcurement customers can use the Oracle Exchange site for their punchouts. See https://exchange.oracle.com/. Look for information about catalogs and Oracle iProcurement customers.

Only the models that use Oracle Exchange require the buyer and supplier to join Exchange. If the buyer accesses the supplier directly, Exchange is not required.

Checklists for Setting Up Punchout

In the following tables, Review Required means you should read the step to see if it applies to you.

Use the following table as your checklist if you are implementing Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML):

Table 2–1   Supplier Setup Steps for Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)

Step Number	Step	Required?
1	Register Company on Oracle Exchange on page 2-9	Required
2	Load Catalog Items to Oracle Exchange on page 2-10	Required

Use the following table as your checklist if you are implementing Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML).

Table 2–2   Supplier Setup Steps for Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)

Step Number	Step	Required?
1	Install XML Parser on Supplier Site on page 2-13	Required
2	Create URL on Supplier Site to Accept Incoming Documents on page 2-14	Required
3	Process Incoming XML or cXML Request on page 2-15	Required
4	Provide XML or cXML Response on page 2-16	Required
5	Determine Method for Returning Buyers to Procurement System on page 2-18	Required
6	Return Shopping Cart on page 2-20	Required
7	Verify Buyer's Access to Secure Site on page 2-31	Required

Use the following table as your checklist if you are implementing Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML):

Table 2–3   Supplier Setup Steps for Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)

Step Number	Step	Required?
1	Register Company on Oracle Exchange on page 2-9	Required
2	Create Data Mappings on Oracle Exchange on page 2-12	Review Required
3	Install XML Parser on Supplier Site on page 2-13	Required
4	Create URL on Supplier Site to Accept Incoming Documents on page 2-14	Required
5	Process Incoming XML or cXML Request on page 2-15	Required
6	Provide XML or cXML Response on page 2-16	Required
7	Determine Method for Returning Buyers to Procurement System on page 2-18	Required
8	Return Shopping Cart on page 2-20	Required
9	Configure Punchout Definition on Oracle Exchange on page 2-23, consisting of the following steps:	—
9-A	■ Configure Punchout Definition (Required) on page 2-25	Required
9-B	■ Test Punchout Definition (Required) on page 2-27	Required
9-C	■ Assign Search Key Words (Optional) on page 2-29	Optional
9-D	■ Enable Punchout Availability (Required) on page 2-30	Required
10	Verify Buyer's Access to Secure Site on page 2-31	Required

Use the following table as your checklist if you are implementing Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML):

Table 2–4   Supplier Setup Steps for Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

Step Number	Step	Required?
1	Register Company on Oracle Exchange on page 2-9	Required
2	Install XML Parser on Supplier Site on page 2-13	Required

Table 2–4   Supplier Setup Steps for Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

Step Number	Step	Required?
3	Create URL on Supplier Site to Accept Incoming Documents on page 2-14	Required
4	Process Incoming XML or cXML Request on page 2-15	Required
5	Provide XML or cXML Response on page 2-16	Required
6	Determine Method for Returning Buyers to Procurement System on page 2-18	Required
7	Return Shopping Cart on page 2-20	Required
8	Configure Punchout Definition on Oracle Exchange on page 2-23, consisting of the following steps:	—
8-A	■ Configure Punchout Definition (Required) on page 2-25	Required
8-B	■ Test Punchout Definition (Required) on page 2-27	Required
8-C	■ Assign Search Key Words (Optional) on page 2-29	Optional
8-D	■ Enable Punchout Availability (Required) on page 2-30	Required
9	Verify Buyer's Access to Secure Site on page 2-31	Required

Use the following table as your checklist if you are implementing Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML):

Table 2–5   Supplier Setup Steps for Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

Step Number	Step	Required?
1	Register Company on Oracle Exchange on page 2-9	Required
2	Create Data Mappings on Oracle Exchange on page 2-12	Optional
3	Install XML Parser on Supplier Site on page 2-13	Required
4	Create URL on Supplier Site to Accept Incoming Documents on page 2-14	Required
5	Process Incoming XML or cXML Request on page 2-15	Required
6	Provide XML or cXML Response on page 2-16	Required

Table 2–5   Supplier Setup Steps for Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

Step Number	Step	Required?
7	Determine Method for Returning Buyers to Procurement System on page 2-18	Required
8	Return Shopping Cart on page 2-20	Required
9	Configure Punchout Definition on Oracle Exchange on page 2-23, consisting of the following steps:	—
9-A	■ Configure Punchout Definition (Required) on page 2-25	Required
9-B	■ Test Punchout Definition (Required) on page 2-27	Required
9-C	■ Assign Search Key Words (Optional) on page 2-29	Optional
9-D	■ Enable Punchout Availability (Required) on page 2-30	Required
10	Verify Buyer's Access to Secure Site on page 2-31	Required

Setting Up Transparent Punchout

This section describes everything the supplier needs to do to set up transparent punchout.

Transparent Punchout Implementation Considerations

In a transparent punchout, the supplier sets up integration with its search engine to properly return search results to Oracle iProcurement. If the supplier has already implemented a punchout, the supplier can leverage some of that setup when implementing a transparent punchout.

Transparent punchout implementation considerations include the following:

- Decide on the model that you and your buyer want to use for the transparent punchout. Which model you use depends on where your catalog content resides (your own site or on Oracle Exchange). See Chapter 1.

- Decide how to validate user access of your site. Look ahead at Process Incoming XML or cXML Request on page 2-15.

- For transparent punchout, suppliers *must* use UTF-8 encoding when sending transparent punchout documents. See Encoding on page C-4 for more information.

- Oracle iProcurement has requirements for item information. Read Appendix B to understand item requirements as described by the document type definitions (DTDs). For example, all items in Oracle iProcurement must have a numeric price; if not, the requester cannot add the item to the cart.

- Consider using the <authenticatedKey> in the ItemSearchResponse XML document. Oracle iProcurement can use this key for subsequent search requests, speeding performance. If your catalog includes a lot of images that you will be returning with the matching items in the search response, using the <authenticatedKey> is recommended. The <authenticatedKey> speeds performance when retrieving images from the database; if your images reside on a Web server, you may not need the <authenticatedKey>. See ItemSearchResponse on page B-50 for more information.

- See ItemSearchRequest on page B-43 and ItemSearchResponse on page B-50 for a complete understanding of the search requirements. For example, to be consistent with Oracle iProcurement behavior, interpret asterisks (*) as wildcards and use AND as the query matching condition among multiple search keywords.

Suppliers of licensed Oracle iProcurement customers can use the Oracle Exchange site for their punchouts. See https://exchange.oracle.com/. Look for information about catalogs and Oracle iProcurement customers.

Only the models that use Oracle Exchange require the buyer and supplier to join Exchange. If the buyer accesses the supplier directly, Exchange is not required.

Checklists for Setting Up Transparent Punchout

Use the following table as your checklist if you are implementing Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML):

Table 2–6   Supplier Setup Steps for Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)

Step Number	Step	Required?
1	Register Company on Oracle Exchange on page 2-9	Required
2	Load Catalog Items to Oracle Exchange on page 2-10	Required

Use the following table as your checklist if you are implementing Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML).

Table 2–7   Supplier Setup Steps for Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)

Step Number	Step	Required?
1	Install XML Parser on Supplier Site on page 2-13	Required
2	Create URL on Supplier Site to Accept Incoming Documents on page 2-14	Required
3	Process Incoming XML or cXML Request on page 2-15	Required
4	Provide XML or cXML Response on page 2-16	Required
5	Verify Buyer's Access to Secure Site on page 2-31	Required

Setup Steps

The following sections describe the supplier setup steps in detail. Refer to the tables in the previous section for the order of the setup steps, depending on the model.

Register Company on Oracle Exchange

Perform this step for the following models:

Model	Required?
Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required
Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)	Required

To enable access to your site or catalog items in these models, you must exist as a registered company on Oracle Exchange. If you are already registered with Oracle Exchange, no additional or special type of registration is necessary.

To register with Oracle Exchange:

1. Access the Exchange with which you wish to participate.

   Oracle Exchange exists at https://exchange.oracle.com/. Oracle Exchange may also exist as a branded Exchange at other sites.

2. Browse the site to see if there are any special requirements or other information for punchout or transparent punchout buyers before registering.

3. Follow the guidance on the Exchange **Home** page to register.

   You can register as an independent entity or as an affiliate.

Load Catalog Items to Oracle Exchange

Perform this step for the following models:

Model	Required?
Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)	Required
Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)	Required

These models typically assume that the supplier loads its catalog items directly to the Oracle Exchange catalog.

To load catalog items to Oracle Exchange:

1. Access Exchange using a signon that has been assigned the Catalog Authoring job function.

   For information on job functions, see the *Company Administration Guide* on Oracle Exchange. To access the guide:

   - Make sure you are logged in as the Company Administrator.

   - Click the "Company Admin" link on the **Home** page.

   - Click the "Download Company Administration Guide" link.

   If you are not assigned this job function, contact the Exchange Operater.

2. Perform the following basic steps to load catalog items to Oracle Exchange:

   - Decide how to categorize and describe catalog items. Understand price lists, including buyer, trading group, and customer group price lists, if you use these.

   - Add catalog items through online or bulk load methods.

   - Set catalog preferences (control whether you want to review your catalog before publishing it).

See the following section of the online Help in Oracle Exchange for complete details on categorizing, describing, and loading catalog items and prices; setting preferences; using third parties to load items for you; and other catalog authoring features:

Online Help: Buying and Selling > Catalog Authoring

Create Data Mappings on Oracle Exchange

Perform this step for the following models:

Model	Required?
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Review Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Optional

Use this step to map the codes that are used in Oracle Exchange to the codes that you use, if they are different. Data mapping allows Oracle Exchange to recognize codes that are passed to it by the supplier or to send codes to the buyer that the buyer's system recognizes. For punchout, the supplier can use Oracle Exchange to map the following codes:

- Currency codes
- Unit of measurement (UOM) codes

This step is required for Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML) if the supplier's currency codes are different from those that Oracle Exchange uses. For the other models, if you perform no mapping on Oracle Exchange, the value is passed as is from Oracle Exchange to the buyer's system. The buyer may have set up data mapping in its system.

For instructions on setting up data mapping on Oracle Exchange, see the *Company Administration Guide* on Oracle Exchange. To access the guide:

- Make sure you are logged in as the Company Administrator.
- Click the "Company Admin" link on the **Home** page.
- Click the "Download Company Administration Guide" link.

On the **Data Mapping** page on Oracle Exchange, enter the Exchange to My Company field:

Description	Exchange Value	My Company to Exchange	Exchange to My Company
Description of the code	The value used internally by Exchange to represent the currency or UOM.	The value the supplier passes to Exchange.	For punchout, the supplier can leave this value as is.

For example, your company may use US as the currency code for *US dollars* while Oracle Exchange uses USD. In this case, you would enter US in the My Company to Exchange field.

Install XML Parser on Supplier Site

Perform this step for the following models:

Model	Required?
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required
Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)	Required

Communication between Oracle Exchange or Oracle iProcurement and a supplier's Web catalog makes extensive use of XML processing. As the supplier, you must have an XML parser tool installed to decode the XML documents passed to you. You can use Oracle XML Parser for Java (version 2.0 or above) or another commercially available parser. (It does not have to be a Java parser.)

Look for Oracle XML Parser software, documentation, and installation instructions at the following URL:

```
http://technet.oracle.com/
```

To download the Apache XML Parser for Java documentation and software, see:

```
http://xml.apache.org
```

Use the instructions on these Web pages when downloading and installing the parser.

Create URL on Supplier Site to Accept Incoming Documents

Perform this step for the following models:

Model	Required?
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required
Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)	Required

To provide the necessary security, it is recommended that you create a page on a secure HTTPS server to handle the request. Although you can use an HTTP server for your Web catalog portal, HTTP does not provide a secure connection. Since password information is passed between Oracle iProcurement (or Oracle Exchange) and your portal, using an unsecure connection may allow the transmission to be intercepted and compromise the security of your site. Digital certificates for the creation of an HTTPS site can be obtained from an established certification authority company. See Appendix C for more details.

How to accept the incoming documents depends on whether you are implementing an XML punchout, cXML punchout, or transparent punchout.

As the supplier, you must create a URL to accept the incoming XML loginRequest or cXML PunchOutSetupRequest document if implementing punchout, or XML ItemSearchRequest document if implementing transparent punchout. Oracle Exchange or Oracle iProcurement will establish a server-to-server communication with this URL and send the request using the HTTP protocol. Your URL will need to receive the HTTP POST document, construct a loginResponse XML document or PunchOutSetupResponse cXML document if implementing punchout, or an

ItemSearchResponse XML document if implementing transparent punchout, and send this response back to Oracle Exchange or Oracle iProcurement (depending on the model).

For XML Punchout

For XML punchout requests, the content type of the HTTP POST in the request header is `application/x-www-form-url-encoded`. This content type means that the login request is URL-encoded, and the supplier must URL-decode the login request.

The parameter name and value pair in the request body consists of the parameter name loginRequest, whose value is the contents of the loginRequest XML document. For example, to retrieve the login request using a Java servlet or JSP, use the following command:

```
String xmlDocument = request.getParameter("loginRequest");
```

For cXML Punchout or Transparent Punchout

For a cXML punchout and transparent punchout requests, the content type of the HTTP POST in the request header is `text/xml`, and the contents of PunchOutSetupRequest and ItemSearchRequest are in the request body. For more information on processing cXML contents, see the *cXML User's Guide* at http://www.cxml.org.

Process Incoming XML or cXML Request

Perform this step for the following models:

Model	Required?
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required
Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)	Required

The XML loginRequest or cXML PunchOutSetupRequest punchout document or the XML ItemSearchRequest transparent punchout document that is sent to the supplier's site contains information about the user and the user's company. As the supplier, you should use this information to validate the proper level of user access.

For examples and descriptions of the XML and cXML punchout and transparent punchout request documents, to see what data you can or need to process, see Appendix B. See also Chapter 3.

When the buyer sets up punchout or transparent punchout access to your site or when you define a punchout on Oracle Exchange, one of the steps is to provide a password for accessing your site. This password is included in the request documents, and it is used by everyone in the accessing application (Oracle iProcurement or Oracle Exchange). You do not need to register individual password accounts on the supplier site. If you want to control or protect access to your site, set up your site to validate this password. Your site may also choose to validate other information included in the request document. For example, your site can check the user name and company name if you are hosting buyer-specific content or price breaks.

Provide XML or cXML Response

Perform this step for the following models:

Model	Required?
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required
Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)	Required

Once the login request is received, the supplier's portal should respond to Oracle iProcurement or Oracle Exchange with either an XML loginResponse or cXML PunchOutSetupResponse document for a punchout or with an XML ItemSearchResponse document for a transparent punchout.

For examples and descriptions of the XML and cXML punchout and transparent punchout response documents, which include both failed and successful responses, see Appendix B.

Errors

If the request is refused access to your Web catalog, the buyer will be shown an error message and will be unable to access the catalog. Oracle iProcurement or Oracle Exchange displays the message. Descriptions of errors are provided in the *Oracle Procurement Buyer's Guide to Punchout and Transparent Punchout*.

For Punchout   If the supplier's site finds an error (for example, the password is invalid), the site should return to Oracle iProcurement or Oracle Exchange a loginResponse XML or PunchOutSetupResponse cXML document indicating an error. See examples in Appendix B. (Once the supplier site sends the response and redirects the buyer to the supplier URL, any errors that occur are displayed by the supplier's system.)

For Transparent Punchout   If the supplier's site finds an error (for example, the password is invalid), the site should return to Oracle iProcurement or Oracle Exchange an ItemSearchResponse XML document indicating an error. See examples in Appendix B.

Punchout Response

 The buying company's setup for the punchout typically creates one logon user account—a single proxy user name or password that is used by all requesters in the company to access the supplier site. Therefore, it is recommended that the supplier develop a method that allows multiple requesters with the same login account to access its site at one time. This method is also known as making the site *session-aware*. This method ensures that the buyer's connection is secure because the session closes as soon as the buyer leaves the supplier's site. The session will also be used to identify the buyer as having navigated to the supplier's site from Oracle iProcurement or Oracle Exchange.

In the punchout response document examples in Appendix B, the session is identified with a session key. Alternatively, the supplier can use some other method for distinguishing logons to its site. For example, the supplier can use an additional unique identifier sent with the user's information to identify the user.

> **Note:** Whatever method the supplier uses to create the session key, the supplier must return the information in the URL in the response document. Oracle iProcurement and Oracle Exchange honor only the information in the URL (<loginURL> in XML or <URL> in cXML). If the supplier does not specify the session information in the URL, then Oracle iProcurement (and Oracle Exchange) lose the information, resulting in login errors in some scenarios. For example, session information that is stored in a cookie is not used by Oracle iProcurement. Suppliers should place the session information solely within the URL.

Transparent Punchout Response

For transparent punchout, the supplier can optionally return an authenticated key in the ItemSearchResponse document using the <AuthenticatedKey> field. When Oracle iProcurement receives this key, it includes the key in every subsequent search request to this supplier. The supplier site can validate the key against the key it previously sent and therefore does not need to authenticate the user again. Use of the authenticated key is recommended because it can improve the performance of the transparent punchout.

Determine Method for Returning Buyers to Procurement System

Perform this step for the following models:

Model	Required?
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required

Once buyers have completed browsing the supplier's site, they must return to Oracle iProcurement or Oracle Exchange (wherever they initiated the punchout) to complete the purchasing process for the items they added to their shopping cart from the supplier's Web catalog.

Depending on the structure of the supplier's Web catalog, the supplier may need to determine whether a requester navigated to its site from Oracle iProcurement. (For a punchout from Oracle Exchange, the supplier would determine whether the buyer navigated to its site from Oracle Exchange.) For example, requesters from Oracle iProcurement should see a Return or Done Shopping button (or however the supplier wishes to name it) instead of the supplier's standard button for completing an order on the supplier's site. This button should return the requester to Oracle iProcurement. Alternatively, the supplier's standard Checkout or Complete Order button should detect where the buyer came from, and return the buyer to Oracle iProcurement.

The supplier may need to create an additional Return button for buyers who have not placed the supplier's items in the shopping cart. In this case, the buyer is not in the cart. Therefore, the supplier can create a Return button on the supplier's shopping pages (outside the cart) to help the buyer return to Oracle iProcurement. In this case, return the buyer to Oracle iProcurement without a cart.

Creating a Return Button for the Shopping Cart

When the buyer is ready to check out the items, the example shown in the next step, demonstrates the output your Return button needs to send to the buyer's browser to successfully return the shopping cart items to the buyer's system.

Creating a Return Button Outside the Shopping Cart

To return the buyer to Oracle iProcurement or Oracle Exchange, create a link using the return URL.

For example, a return URL to Oracle iProcurement may look like this:

```
http://qapache.us.oracle.com:15671/OA_HTML/OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_
CALLBACK&OAHP=ICX_POR_HOMEPAGE_MENU&OASF=ICX_CAT_PUNCHOUT_
CALLBACK&transactionid=1577779317
```

A return URL to Oracle Exchange may look like this:

```
https://exchange.oracle.com/orders/PunchoutCallBack.jsp
```

Since the buyer did not purchase items from the supplier site, the URL does not contain shopping cart information, and the buyer is returned to the buying system.

When constructing the return button, the supplier can either link the button directly to the return URL as shown above, or create code that directs the buyer to the supplier's server, which then returns the buyer to the buyer's system.

Return Shopping Cart

Perform this step for the following models:

Model	Required?
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required

Once the buyer clicks the Return button, your site must send a transaction, via the buyer's browser, to Oracle iProcurement or Oracle Exchange containing details of all the items the buyer selected from your Web catalog.

For Oracle native XML, the URL to which the XML shoppingCart document should be sent is the URL identified by the <returnURL> tag in the original XML loginRequest document. The parameter name for the HTTP POST is oracleCart, and the cart should be URL-encoded.

For cXML, the URL to which the cXML shopping cart document (PunchOutOrderMessage) should be sent is identified by the <BrowserFormPost> tag in the original cXML PunchOutSetup Request. See the *cXML User's Guide* at http://www.cxml.org for more information. For cXML, the parameter name for the HTTP POST is either cXML-urlencoded or cXML-base64. The cXML-urlencoded parameter should be URL-encoded. The cXML-base64 parameter should be Base64-encoded.

Once the document is returned to Oracle iProcurement or Oracle Exchange, the selected items are added to the buyer's shopping cart, and the buyer can complete the purchasing process.

In addition, suppliers should determine whether to return configuration numbers (in addition to item numbers) in the shopping cart to Oracle iProcurement. If so, suppliers should decide whether to allow re-punchout so that requesters can examine the configuration again later (for example, during requisition approval). Re-punchout is available in cXML only. See Configurations and Re-Punchout on page 1-16.

For examples and descriptions of the XML and cXML shopping cart documents, see Appendix B.

The following example shows how the XML shopping cart information should be sent back to Oracle iProcurement via the requester's browser. (For information on returning cXML shopping carts, see the *cXML User's Guide* at http://www.cxml.org.) In the example, note the following:

- The line `onload="document.orderForm.submit()"` asks the browser to submit itself.

- `ACTION="..."` is the Oracle iProcurement URL to which the cart is posted.

- The value of the parameter oracleCart is the url-encoded shopping cart data.

- The + signs in the example below represent spaces in the shopping cart document, demonstrating that the cart is url-encoded. The spaces are created by calling URLEncoder.encode on the shopping cart XML. URLEncoder can be found in the java.net package. (The java.net package handles ISO-8859-1 characters only. If multibyte characters are used, look for or write your own URL encoder to handle these.)

The following example returns shopping cart items to Oracle iProcurement:

```
<HTML>
<BODY onload="document.orderForm.submit()">
<FORM

ACTION="http://qapache.us.oracle.com:15671/OA_HTML/OA.jsp?OAFunc=ICX_CAT_
PUNCHOUT_CALLBACK&OAHP=ICX_POR_HOMEPAGE_MENU&OASF=ICX_CAT_PUNCHOUT_
CALLBACK&transactionid=1577779317"
METHOD="POST" NAME="orderForm">
<INPUT TYPE="HIDDEN" NAME="oracleCart"

VALUE="%3C%3Fxml+version+%3D+'1.0'+encoding+%3D+'UTF-8'%3F%3E%0A%3Cresp
onse%3E%0A+++%3Cheader%3E%0A++++++%3Caction%3E%0A+++++++++%3C!%5BCDATA%5BSHO
PPING_RESPONSE%5D%5D%3E%0A++++++%3C%2Faction%3E%0A+++%3C%2Fheader%3E%0A+++%3
Cbody%3E%0A++++++%3COrderLinesDataElements%3E%0A+++++++++%3CcatalogTradingPa
rtner%3E%0A+++++++++++%3C!%5BCDATA%5BOracleExchange%5D%5D%3E%0A+++++++++%3C
%2FcatalogTradingPartner%3E%0A+++++++++%3COrderLine%3E%0A++++++++++++%3Ccont
ract%3E%0A++++++++++++++%3CcatalogType%3E%0A++++++++++++++++++%3C!%5BCDATA%
5BCONTRACTED%5D%5D%3E%0A++++++++++++++%3C%2FcatalogType%3E%0A++++++++++++%3
C%2Fcontract%3E%0A++++++++++++%3Citem+lineType%3D%22GOODS%22+quantity%3D%221
.0%22%3E%0A++++++++++++++%3CitemNumber%3E%0A++++++++++++++++++%3CsupplierIt
emNumber%3E%0A++++++++++++++++++++%3CitemID%3E%0A++++++++++++++++++++++++%3
C!%5BCDATA%5B6ZC75%5D%5D%3E%0A++++++++++++++++++++%3C%2FitemID%3E%0A+++++++
```

```
+++++++++++%3C%2FsupplierItemNumber%3E%0A++++++++++++++%3C%2FitemNumber%3E%
0A+++++++++++++++%3CitemDescription%3E%0A++++++++++++++++++%3C!%5BCDATA%5BGe
neral+Duty+Engraving+Pen%2C+9000+RPM+Rotational+Speed%2C+6-5%2F8+inch+Length
%5D%5D%3E%0A+++++++++++++++%3C%2FitemDescription%3E%0A+++++++++++++++%3Cunit
OfMeasure%3E%0A+++++++++++++++++%3CbuyerUnitOfMeasure%3E%0A+++++++++++++++++
+++++%3CbuyerUOMType%3E%0A+++++++++++++++++++++++%3C!%5BCDATA%5BEach%5D%5D%
3E%0A+++++++++++++++++++++%3C%2FbuyerUOMType%3E%0A++++++++++++++++++%3C%2Fbu
yerUnitOfMeasure%3E%0A++++++++++++++++++%3CsupplierUnitOfMeasure%3E%0A++++++
+++++++++++++++%3CsupplierUOMType%3E%0A++++++++++++++++++++++++++%3C!%5BCDATA%
5BEA%5D%5D%3E%0A+++++++++++++++++++++%3C%2FsupplierUOMType%3E%0A+++++++++++++
+++++++++%3CsupplierUOMQuantity%3E%0A+++++++++++++++++++++++%3C!%5BCDATA%5B
%5D%5D%3E%0A++++++++++++++++++++%3C%2FsupplierUOMQuantity%3E%0A+++++++++++++
++++++%3C%2FsupplierUnitOfMeasure%3E%0A+++++++++++++++%3C%2FunitOfMeasure%3E
%0A++++++++++++%3C%2Fitem%3E%0A++++++++++++%3Ccategory%3E%0A+++++++++++++++%
3CcategoryCode+categoryCodeIdentifier%3D%22SUPPLIER%22%3E%0A++++++++++++++++
++%3C!%5BCDATA%5BImpact+wrenches%5D%5D%3E%0A+++++++++++++++%3C%2FcategoryCod
e%3E%0A+++++++++++%3C%2Fcategory%3E%0A++++++++++++%3Cprice%3E%0A++++++++++++
++++%3Ccurrency%3E%0A+++++++++++++++++%3C!%5BCDATA%5BUSD%5D%5D%3E%0A+++++++
++++++++%3C%2Fcurrency%3E%0A+++++++++++++++%3CunitPrice%3E%0A+++++++++++++++
+++%3C!%5BCDATA%5B179.08%5D%5D%3E%0A+++++++++++++++%3C%2FunitPrice%3E%0A++++
++++++++%3C%2Fprice%3E%0A++++++++++++%3Csupplier%3E%0A+++++++++++++++%3Csupp
lierDUNS%3E%0A+++++++++++++++++%3C!%5BCDATA%5B005103494%5D%5D%3E%0A++++++++
+++++++%3C%2FsupplierDUNS%3E%0A+++++++++++++++%3CsupplierName%3E%0A+++++++++
+++++++++%3C!%5BCDATA%5BGrainger%5D%5D%3E%0A+++++++++++++++%3C%2FsupplierNam
e%3E%0A+++++++++++++%3C%2Fsupplier%3E%0A+++++++++%3C%2ForderLine%3E%0A++++++%
3C%2FOrderLinesDataElements%3E%0A+++%3C%2Fbody%3E%0A%3C%2Fresponse%3E%0A">
</FORM></BODY></HTML>
```

Return Search Response

Perform this step for the following model:

Model	Required?
Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)	Required

Once your site generates the search results, it must send an ItemSearchResponse XML transaction in the HTTP response.

The content type in the response header should be text/xml, and the entire ItemSearchResponse document is contained in the body of the HTTP response. The document should be encoded in UTF-8 or, if the contents of the document are all ASCII characters, in ASCII.

For examples and descriptions of the XML ItemSearchResponse document, including how to optionally return images of items, see Appendix B.

Configure Punchout Definition on Oracle Exchange

Perform this step for the following models:

Model	Required?
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required

In this step, you use the **Configure Catalog Punchout** page to define and test your punchout definition on Oracle Exchange so that buyers can access it. This step consists of the following tasks:

1. Configure punchout definition

2. Test punchout definition

3. Assign search keywords

4. Control availability

The following illustration shows the **Configure Catalog Punchout** page that you will use to perform these tasks:

Welcome   **Catalogs**   Negotiations   Sales   Intelligence

Catalogs > Configure Catalog Punchout

Configure Catalog Punchout

Step 1: Configure Punchout Definition

Click here to configure your punchout on the Oracle Exchange. Before you undertake this step, you must have first enabled your web site to support catalog punchout from the Oracle Exchange. To learn more about enabling catalog punchouts, click here to download the punchout documentation.

Step 2: Test Punchout Definition

Before you continue, you should test your punchout to ensure that this functions as expected. Click here to test your punchout to you web catalog.

Step 3: Assign Search Keywords

In order for your punchout link to appear in searches which buyer's execute on the Oracle Exchange, you need to define keywords for your punchout. Click here to define keywords for your punchout.

Step 4: Control Availability

To make your punchout available to buyers on the Oracle Exchange, you must publish your punchout. After you have published your punchout, you can revoke it at any time. Click here to set the availability of your punchout.

Prerequisites

You must be assigned the following Oracle Exchange system tasks to complete the punchout definition:

- Catalog Authoring - Configure Catalog Punchout—typically contained in the Catalog Authoring job function, unless the Exchange Operator modified this job function. This system task is needed to perform all of the punchout configuration setup in this section.

- Create Add-to-Cart Order—typically contained in a Buyer job function, unless the Exchange Operator modified the job function. This system task is needed to perform the Test Punchout Definition step, where you will shop and add your site's punchout items to the cart.

The Exchange Operator assigned job functions when approving your registration. For information on job functions, see the *Company Administration Guide* on Oracle Exchange. To access the guide:

- Make sure you are logged in as the Company Administrator.

- Click the "Company Admin" link on the **Home** page.

- Click the "Download Company Administration Guide" link.

If you are not assigned the proper job functions that contain these system tasks, contact the Exchange Operater.

Configure Punchout Definition (Required)

To configure your punchout definition:

1. Access Exchange with a signon that has been assigned the Catalog Authoring - Configure Catalog Punchout system task. (See Prerequisites, above.)

2. On the **Home** page, click the "Selling" link, then the Catalogs tab.

3. On the **Catalog Authoring** page, click the "Configure Catalog Punchout" link.

4. On the **Configure Catalog Punchout** page, click the "Click here to configure your punchout" link.

5. On the **Configure Punchout Definition** page, enter values in the Punchout URL and Punchout Password fields:

   - Punchout URL. Address of your Web catalog's portal page. This is the URL you created in a prior step and is the address Oracle Exchange will contact when processing a punchout request. For example: https://ap999sun.us.oracle.com:6666/orders/testscripts/SupplierLogin.jsp

   - Punchout Password. The value that Oracle Exchange passes to your Web catalog in the login request document when a buyer accesses your catalog. Your site should validate this password, if you wish to validate the password before allowing access. This is the password that every buyer uses to access your site; you do not need to create separate password accounts for every buyer.

| Welcome | Catalogs | Negotiations | Sales | Intelligence |

Catalogs > Configure Catalog Punchout > Configure Punchout Definition

Configure Punchout Definition

Enter your Punchout configuration and the press the **Apply Changes** button below when you're done.

* Indicates a required field

Getting Started

Enabling punchout means buyers can shop directly at your site and add items to the Oracle Exchange shopping cart.

Want to learn more about how this works and set it up?

Download Punchout Documentation

Access

The punchout URL and password are used to make the connection between Oracle Exchange and your punchout catalog.

* Punchout URL `http://ap995sun.us.oracle.com:6245/orders/testscripts/SupplierLogin.jsp`
(Example: http://www.mysite.com)

* Punchout Password `*******`

Communication Protocol

Decide which communication protocol is to be used by your punchout. For more information on communication protocols download the Punchout Documentation.

⦿ XML
◯ cXML

User Interface Identification

Decide how you want to present your company in the catalog. Both the business description and logo are displayed next to your name in the listing of punchout enabled suppliers. Display name is shown whenever your company is referenced.

Display Name `Gorilla`
Enter only if you don't want to use your registered company name

Logo URL `[                              ]`
(Example: http://www.mysite.com/logo.gif)

If you provide a logo, it cannot exceed 150 (width) X 50 (height) pixels.

Company Description `[                              ]`
(Example: The largest online retailer of office supplies and modular furniture.)

Apply Changes

6. Select the communication protocol you want Oracle Exchange to use to send punchout information to you. Oracle Exchange supports both XML and cXML, depending on the model being implemented.

7. You may optionally enter the following additional information in the User Interface Identification section. This information displays to the buyer with your punchout link.

   - You can provide an alternate company name in the Display Name field. This name will be used in place of your company's registered name.

   - You can supply the location of a company logo image in the Logo URL field. This image must be in .gif format and no larger than 125 pixels wide and 25 pixels high. This URL address must be outside your firewall so Oracle Exchange can access the image.

   - You can add a brief description of your catalog and the items you sell in the Company Description field.

   > **Note:** To provide a name or description in other Oracle Exchange languages, return to the Oracle Exchange **Home** page, click a language to select it (if Oracle Exchange is available in multiple languages, other languages display beneath the tabs), and return to the **Configure Punchout Definition** page to provide the name or description in that language. If you do not provide the name or description in the other languages, Oracle Exchange displays the same name or description you first entered in all languages. (If you left the name blank, Oracle Exchange assumes the registered company name.)

8. Click Apply Changes.

9. On the Confirmation page, click the "Return to Configure Catalog Punchout" link to proceed to testing your punchout definition.

Test Punchout Definition (Required)

If you receive an error message at any point in the following steps, make a note of the message to assist in resolving the problem.

To test your punchout definition:

1. Access Exchange with a signon that has been assigned the Catalog Authoring - Configure Catalog Punchout and Create Add-to-Cart Order system tasks. (See Prerequisites, above.)

2. On the **Home** page, click the "Selling" link, then the Catalogs tab.

3. On the **Catalog Authoring** page, click the "Configure Catalog Punchout" link.

4. On the **Configure Catalog Punchout** page, click the "Click here to test your punchout" link.

5. On the **Test Catalog Punchout** page, click Begin Punchout Test Now.

| Welcome | Catalogs | Negotiations | Sales | Intelligence |

Catalogs > Configure Catalog Punchout > Test Catalog Punchout

Test Catalog Punchout

Clicking the **Begin Punchout Test Now** button will initiate the punchout to your web site as if you were a buyer. You should verify that you can select items from your webstore, and that these can be successfully returned to the Oracle Exchange.

Before you begin this process you must have configured your website to support catalog punchouts. To learn more about configuring your website for catalog punchouts, click here to download detailed documentation.

When you have finished testing your punchout, you will need to return to the Configure Catalog Punchout page, and complete the punchout configuration process by assigning search keywords and making your punchout available to registered buyers.

® **Note**: in order to return to the System with your test shopping cart, you must be assigned the "Create Add-to-Cart Order" task. If you do not have this task assigned, you will receive an error when you attempt to return your shopping cart to the System. Contact your System Operator to have this task assigned to you.

Begin Punchout Test Now

6. If your definition has been correctly defined, you will be able to punch out to your site and select items from your catalog.

   If the punchout does not work, Oracle Exchange returns an **Error** page. Click the linked error to see the detailed **Error** page, to help you determine the cause of the error. Also see Verify Buyer's Access to Secure Site on page 2-31.

7. Once you have selected several items, return to Oracle Exchange by clicking the Return to Exchange button (or whatever name you used when creating your redirect earlier).

8. Verify your shopping cart contents have been successfully added to the Oracle Exchange shopping cart.

9. Select the items and click Delete to empty the cart so you do not create a purchase order accidentally.

Assign Search Key Words (Optional)

Defining appropriate search keywords for items in your Web catalog is important to ensure that buyers browsing Oracle Exchange are presented with the punchout link to your catalog. If a buyer's search criteria include at least one of the keywords you define, your Web catalog will be displayed in the list of suppliers presented to the buyer on the **Search Results** page.

Keywords define the items that you have available on your Web catalog, and you can enter up to 4,000 bytes of information for each language for which you require keywords. For example, if you sell paper products to both English and French speaking customers, you may want to include the following keywords in American English:

  ream paper photocopy Xerox A5 A4 A3 A2 A1 Letter

and the following keywords in French:

  rame papier enveloppe photocopie A5 A4 A3 A2 A1

If a buyer searches on Oracle Exchange in a foreign language for which you have defined no keywords, no punchout link to your catalog will be displayed in the search results. In the example above, if the buyer searches for *papel* in Spanish, a link to your catalog is not returned because you defined no keywords for Spanish.

To create search keywords on Oracle Exchange:

1. Access Exchange with a signon that has been assigned the Catalog Authoring - Configure Catalog Punchout system task. (See Prerequisites, above.)

2. On the **Home** page, click the "Selling" link, then the Catalogs tab.

3. On the **Catalog Authoring** page, click the "Configure Catalog Punchout" link.

4. On the **Configure Catalog Punchout** page, click the "Click here to define keywords for your punchout" link.

5. On the **Trading Partner Keywords** page, select the Language for which you wish to define keywords.

6. Enter all appropriate key words in the text area, separating each word with a space.

7. Click Apply Changes.

8. Repeat these steps for any additional languages your Web catalog supports.

Enable Punchout Availability (Required)

Your punchout definition is not initially available to buyers. You must publish your punchout definition before buyers can use it.

To publish your punchout definition:

1. Access Exchange with a signon that has been assigned the Catalog Authoring - Configure Catalog Punchout system task. (See Prerequisites, above.)

2. On the **Home** page, click the "Selling" link, then the Catalogs tab.

3. On the **Catalog Authoring** page, click the "Configure Catalog Punchout" link.

4. On the **Configure Catalog Punchout** page, click the "Click here to set the availability of your punchout" link.

5. On the **Control Punchout Availability** page, select "Yes, publish my punchout to registered buying companies."

6. Click Apply Changes.

   Your punchout is now available to buyers. Buyers still cannot access the punchout, however, until the buying company's Exchange Purchasing Manager chooses to display the punchout to buyers in the company.

As the supplier, you can revoke your punchout definition at any time by clicking "No, I do not want to publish my punchout to buying companies at this time" on the **Control Punchout Availability** page. The punchout will immediately be unavailable for use by any buyer. The punchout definition itself remains on Oracle Exchange, but cannot be seen and used by buyers. The Purchasing Manager (the buying company's administrator) on Oracle Exchange can still see the punchout definition when choosing which definitions to display to its own buyers, but Oracle Exchange will clearly identify the definition as having been disabled by you. (If the Purchasing Manager disallows its buyers' access to the punchout and saves this change, the punchout link is not displayed to the Purchasing Manager again until you re-publish the punchout.)

Verify Buyer's Access to Secure Site

Perform this step for the following models, if your site is secure (if the site URL begins with HTTPS):

Model	Required?
Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)	Required
Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Required
Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)	Required
Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)	Required
Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)	Required

Typically, the supplier does not need to do anything special to verify that the buyer can access the supplier's secure site; however, the following points summarize what is required for each model when the supplier's site is secure:

- For Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML), the supplier's certification authority needs to be on the ca-bundle.crt file in Oracle iProcurement. The buyer is asked to make sure of this in the *Oracle Procurement Buyer's Guide to Punchout and Transparent Punchout*.

- For Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML), the supplier's certification authority needs to be on the pomdigcrt.txt file in Oracle Exchange.

- For Models 4 and 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML or cXML), the supplier's certification authority needs to be on the pomdigcrt.txt file in Oracle Exchange.

- For Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML), the supplier's certification authority needs to be on the ca-bundle.crt file in Oracle iProcurement. The buyer is asked to make sure of this in the *Oracle Procurement Buyer's Guide to Punchout and Transparent Punchout*.

Both ca-bundle.crt and pomdigcrt.txt come with various certification authorities. If the supplier's certification authority is a common one, it may already be on the ca-bundle.crt and pomdigcrt.txt files.

Contact your network administrator for more information. See also Appendix C.

3

Controlling Access to Punchout or Transparent Punchout

This chapter includes the following topics:

- Buyer Control on page 3-1
- Supplier Control on page 3-2
- Buyer and Supplier Control on page 3-3

This chapter discusses the options that buyers and suppliers have for controlling requesters' access to the punchout or transparent punchout catalogs.

Buyer Control

The buyer can use any of the following methods to control access to a punchout or transparent punchout:

Realms

Buyers in Oracle iProcurement can use realms to control access to a punchout or transparent punchout. Realms restrict access by user or by responsibility.

The buying company can use realms for the following models:

- Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)
- Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML)
- Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML)

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

- Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)

- Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)

Instructions for setting up realms are discussed in the buyer setup.

Control Punchout Access on Oracle Exchange

The buying company can hide or display punchout catalogs using the **Control Punchout Access** page on Oracle Exchange. This page hides or displays the punchout catalog for all registered Oracle Exchange users in the buying company.

The buying company can use the **Control Punchout Access** page in the following models:

- Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

Instructions are discussed in the buyer setup on controlling punchout access on Oracle Exchange.

Supplier Control

The supplier can use any of the following methods to control access to a punchout or transparent punchout:

Authentication of XML or cXML Request

The supplier can use certain fields in the XML or cXML request to authenticate the requester. Some fields never change in a single punchout or transparent punchout,

and can be counted on to always be the same. Others vary by user. A list of these fields is provided in the appendix, for each of the following documents:

- loginRequest (XML Punchout)
- PunchOutSetupRequest (cXML Punchout)
- ItemSearchRequest (XML Transparent Punchout)

Disable a Punchout on Oracle Exchange

Suppliers can enable or disable their punchout definitions on Oracle Exchange by controlling punchout availability. This feature hides or displays their punchout catalog for anyone who accesses it on or via Oracle Exchange.

Suppliers can enable or disable punchout definitions in the following models:

- Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)
- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)
- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

Instructions are discussed in the supplier setup on configuring the punchout definition.

Buyer and Supplier Control

The following methods involve both buyer and supplier setup.

Multiple Organization Punchout or Transparent Punchout with Oracle Exchange

If the buying company has multiple organizations set up in Oracle Applications, the purchasing administrator can configure Oracle Exchange to mirror the multiple organizations setup.

Figure 3–1 illustrates multiple organizations in a punchout or transparent punchout to Oracle Exchange. Figure 3–2 illustrates multiple organizations in a punchout or transparent via Oracle Exchange to the supplier.

Figure 3–1 Punchout or Transparent Punchout to Oracle Exchange
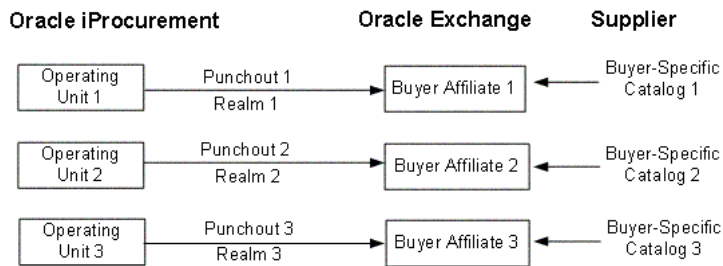
Figure 3–2 Punchout or Transparent Punchout via Oracle Exchange
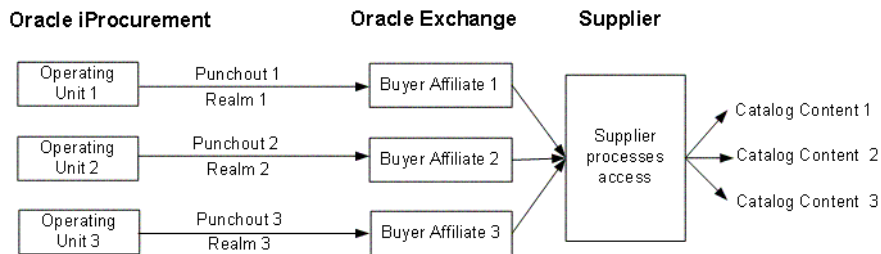
Buyer Affiliates

In the following steps, assume the following scenario: the buying company's multiple organizations that are set up in Oracle Applications include the following two operating units:

- Vision Corp. Germany

- Vision Corp. USA

The supplier Acme 1 supplies goods only to Vision Corp. USA, and the supplier Acme 5 supplies goods only to Vision Corp. Germany. The buying company wants requesters (users) associated with the operating unit Vision Corp. Germany to see content only from Acme 5 when accessing Oracle Exchange. Users associated with the operating unit Vision Corp. USA should see content only from Acme 1.

Exchange Setup To accomplish a multiple organizations setup to or via Oracle Exchange, the buying company should perform the following steps on Oracle Exchange:

1. When you register your buying company on Oracle Exchange, register two companies—one for Vision Corp. Germany and one for Vision Corp. USA.

2. When you register users for your buying company on Oracle Exchange, register a proxy user for Vision Corp. Germany (you could use the user name USApunchout) and another proxy user for Vision Corp. USA (you could use the user name GERpunchout).

Oracle Applications Setup The buying company should perform the following steps in Oracle Applications and Oracle iProcurement:

1. Create a responsibility called (for example) Requisitioner USA.

2. For the Requisitioner USA responsibility, set the profile option *MO: Operating Unit* to the operating unit Vision Corp. USA.

   This step associates the Requisitioner USA responsibility with the Vision Corp. USA operating unit.

3. Create your punchout or transparent punchout for Vision Corp. USA, following the instructions for defining the punchout or transparent punchout.

   You could name the punchout or transparent punchout catalog Exchange - USA. The user name and password associated with this catalog should be the same as the user name and password associated with the user you registered under Vision Corp. USA on Oracle Exchange.

4. Create a realm for the catalog (Exchange - USA) created in the previous step.

5. Assign this realm to the responsibility Requisitioner USA.

6. Repeat these steps for the operating unit Vision Corp. Germany.

When all of these steps are completed, the following data would exist:

Oracle Applications Operating Unit	Responsibility	Catalog	Exchange Company	Exchange User
Vision Corp. USA	Requisitioner USA	Exchange - USA	Vision Corp. USA	USApunchout
Vision Corp. Germany	Requisitioner GER	Exchange - GER	Vision Corp. Germany	GERpunchout

For a punchout or transparent punchout to Oracle Exchange When the supplier loads catalog items and pricing to the Oracle Exchange catalog, it can specify prices specifically for a buying company on Oracle Exchange. For example, Acme 5 will load prices that are visible only to Vision Corp. Germany, and Acme 1 will load prices that are visible only to Vision Corp. USA. Alternatively, a single supplier can load buyer prices that are visible only to Vision Corp. Germany and buyer prices that are visible only to Vision Corp. USA.

Furthermore, because of the realms configuration, the catalog Exchange - USA is available only to a requester who is assigned the responsibility Requisitioner USA. This requester has access only to catalog content that the supplier loaded for the company Vision Corp. USA. Similarly, the catalog Exchange - GER is available only to a requester who is assigned the responsibility Requisitioner GER. This requester has access only to catalog content that the supplier loaded for Vision Corp. Germany.

For a punchout or transparent punchout via Oracle Exchange The supplier creates specific catalogs at its site. The supplier site looks at the buyer affiliate company information sent in the login request. (For example, the supplier could process the CompanyName or From Credential.) Based on the company information, the supplier site determines whether to display one catalog or the other. For example, assume one supplier, Acme. If the company information indicates that the punchout request is coming from Vision Corp. USA, then Acme displays its Catalog Content A. If the punchout request is coming from Vision Corp. Germany, then Acme displays its Catalog Content B.

Furthermore, because of the realms configuration, the catalog Exchange - USA is available only to a requester who is assigned the responsibility Requisitioner USA. This requester has access only to catalog content that the supplier displays for the company Vision Corp. USA. Similarly, the catalog Exchange - GER is available only to a requester who is assigned the responsibility Requisitioner GER. This requester has access only to catalog content that the supplier displays for Vision Corp. Germany.

Supplier Affiliates
The supplier can also create affiliates on Oracle Exchange:

Table 3–1   Supplier Affiliates

Model	Buyer Action	Supplier Action
For Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML),	Buying company creates a punchout to each supplier affiliate.	Each supplier affiliate loads its own catalog.
For Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)	Buyer on Oracle Exchange accesses a different punchout for each supplier affiliate.	Each supplier affiliate creates its own punchout on Oracle Exchange.
For Models 4 and 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML or cXML)	Buying company downloads a punchout for each supplier affiliate.	Each supplier affiliate creates its own punchout on Oracle Exchange. Supplier must process each punchout as desired.
Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)	Buying company creates a transparent punchout to each supplier affiliate.	Each supplier affiliate loads its own catalog.

Multiple Organization Punchout or Transparent Punchout to Supplier

If the buying company has multiple organizations set up in Oracle Applications, the purchasing administrator may want the supplier to distinguish the requesters based on the operating unit to which they belong. For example, only requesters associated with a particular operating unit can view catalog content relevant to that operating unit.

Figure 3–3 illustrates multiple organizations in a punchout or transparent punchout directly to the supplier.

Figure 3–3   Punchout or Transparent Punchout to Supplier
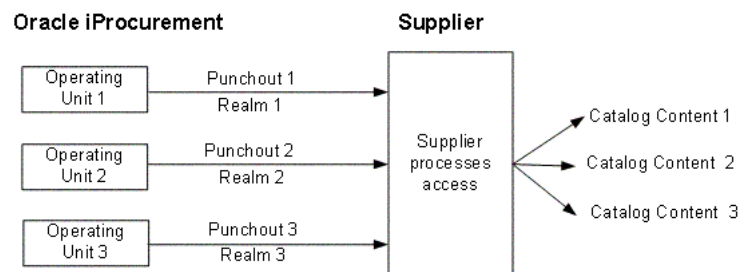
Setup

Assume the following scenario: the buying company's multiple organizations that are set up in Oracle Applications include the following two operating units:

- Vision Corp. Germany
- Vision Corp. USA

Requesters (users) associated with the operating unit Vision Corp. Germany have access only to specific catalog content at the supplier site. Users associated with the operating unit Vision Corp. USA have access to different catalog content than requesters in Vision Corp. German.

To accomplish a multiple organizations punchout or transparent punchout to the supplier, the buying company should perform the following steps in Oracle Applications and Oracle iProcurement:

1. Create a responsibility called (for example) Requisitioner USA and another called (for example) Requisitioner GER.

2. For the Requisitioner USA responsibility, set the profile option *MO: Operating Unit* to the operating unit Vision Corp. USA.

    This step associates the Requisitioner USA responsibility with the Vision Corp. USA operating unit.

3. For the Requisitioner GER responsibility, set the profile option *MO: Operating Unit* to the operating unit Vision Corp. Germany.

    This step associates the Requisitioner GER responsibility with the Vision Corp. Germany operating unit.

4. Set up two punchout catalogs (or two transparent punchout catalogs) to the same supplier site, following the instructions for defining the punchout or transparent punchout.

    For example, create two punchout catalogs with the following names:

    - Supplier Germany
    - Supplier USA

5. For each punchout (or transparent punchout) setup, enter different values in the Identity field.

    For example:

    - For Supplier Germany, enter an Identity of VisionGermany.

- For Supplier USA, enter an Identity of VisionUSA.

For an XML punchout, enter different values in the Company Name or Company ID field.

The supplier may additionally require the buying company to enter a different Password for each punchout (or transparent punchout) setup.

6. Create a realm for the catalog Supplier USA and assign this realm to the responsibility Requisitioner USA.

   This step ensures that only people using the Requisitioner USA responsibility can see the punchout.

7. Create a realm for the catalog Supplier Germany and assign the realm to the responsibility Requisitioner GER.

When all of these steps are completed, the following data would exist:

Oracle Applications Operating Unit	Responsibility	Catalog	Identity (cXML) or Company Name (XML)
Vision Corp. USA	Requisitioner USA	SupplierUSA	VisionUSA
Vision Corp. Germany	Requisitioner GER	SupplierGermany	VisionGermany

Because of the realms configuration, the requester in Vision Corp. USA, who logs into Oracle iProcurement using the Requisitioner USA responsibility, sees only the punchout catalog SupplierUSA. The requester in Vision Corp. Germany, who logs in using the Requisitioner GER responsibility sees only the punchout catalog SupplierGermany. The supplier uses the value in the appropriate field (for example, the Identity field) to determine which operating unit the requester belongs to and presents only the content relevant to that requester (or to anyone from that operating unit with access to the punchout).

A

Detailed Punchout and Transparent Punchout Process

This appendix expands on Chapter 1 by describing the punchout process in more detail for each model:

- Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML) on page A-1

- Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML) on page A-4

- Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML) on page A-7

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML) on page A-10

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML) on page A-13

- Models 1 and 2: Transparent Punchout to Oracle Exchange or Supplier (XML) on page A-17

Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)

The following diagram shows the flow for Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML).

Figure A–1   Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML)
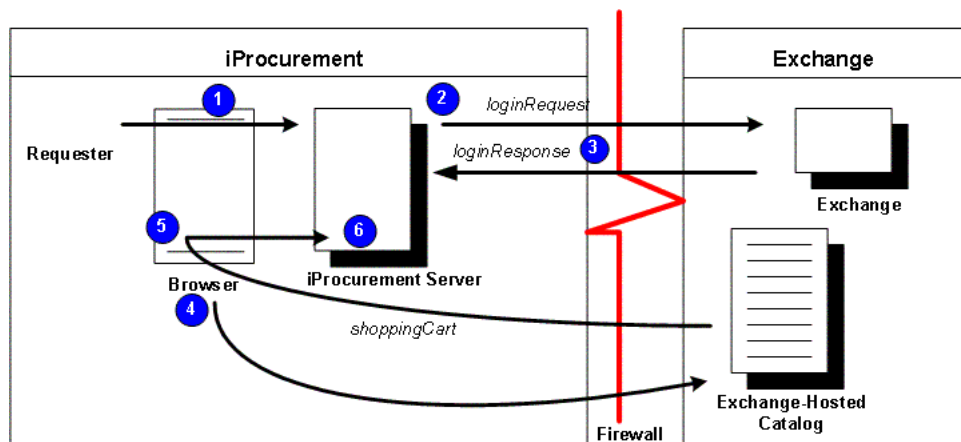
The following sections describe the steps in Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML) in detail.

1 Requester logs on to Oracle iProcurement

The requester logs on to Oracle iProcurement.

2 Requester clicks punchout link to Oracle Exchange

From Oracle iProcurement, the requester clicks a punchout link to Oracle Exchange. When the requester clicks the link, the browser connects to the Oracle iProcurement server.

The Oracle iProcurement server first calls Oracle secure socket layer (SSL) application programmable interfaces (APIs) to establish a secure connection with Oracle Exchange, if Oracle Exchange is secure. (Exchange.Oracle.com is a secure site.) The APIs request the Certification Authority (CA) digital certificate from Oracle Exchange. This digital certificate is compared to certificate authorities stored in the ca-bundle.crt file in Oracle iProcurement. (See Appendix C for more information on how secure connections work.)

After the secure connection is established, the Oracle iProcurement server generates the *loginRequest* XML document, which includes a base set of user details and the return URL for the Oracle iProcurement instance. Oracle iProcurement passes this XML document to the Oracle Exchange site in an HTTP request using the POST format.

3 Oracle Exchange authenticates requester and returns response

Oracle Exchange receives the loginRequest and authenticates the Oracle iProcurement requester. (When the buyer sets up punchout access as described in this guide, this logon is invisible to the requester; the requester does not physically log on again to Oracle Exchange.) The user name and password used to authenticate the requester are the same as the proxy user name and password that the Oracle iProcurement administrator created when registering the proxy user on Oracle Exchange.

Once the requester is authenticated, Oracle Exchange generates the *loginResponse* XML document and sends it to Oracle iProcurement in an HTTP response.

Oracle Exchange is *session-aware*. Although individual requesters access Oracle Exchange with the same proxy user account, Oracle Exchange treats each logon as its own session and allows multiple punchout requesters from a single buying company to punch out to Oracle Exchange at the same time. The session key is provided in the <loginURL> tag of the loginResponse document.

4 Oracle iProcurement redirects browser to Oracle Exchange for shopping

Now that a secure and trusted connection is established and the requester has been assigned a session, Oracle iProcurement redirects the requester's browser to the shopping page on Oracle Exchange. There, the requester is allowed to search for and add items to the shopping cart on Oracle Exchange. The Oracle Exchange functionality available to the requester is based on the job functions that the Oracle iProcurement administrator assigned to the proxy user when registering the proxy user on Oracle Exchange.

Oracle Exchange detects that the shopper is from Oracle iProcurement and provides a Done Shopping button when the requester is finished adding items to the shopping cart. (If the buyer has set up data mapping on Oracle Exchange, Exchange performs the data mapping when the buyer clicks Done Shopping.)

5 Requester finishes shopping; Oracle Exchange returns cart to Oracle iProcurement

When the requester finishes adding items to the cart on Oracle Exchange and clicks Done Shopping, the Oracle Exchange session is dropped and the requester is logged out of Oracle Exchange. Oracle Exchange also returns the shopping cart to the Oracle iProcurement server via the requester's browser. The return to the requester's browser is invisible to the requester. Specifically, the browser uses the HTML FORM POST action to pass the items in the shopping cart to Oracle

iProcurement using the parameter oracleCart. (The oracleCart parameter contains the entire shopping cart contents.) The cart is contained in a hidden form field. To post the shopping cart items, the FORM POST uses the return URL that Oracle iProcurement sent in the loginRequest document.

Oracle Exchange URL-encodes and returns the shopping cart information to Oracle iProcurement in the *shoppingCart* XML document.

> **Note:** Requesters cannot add To Be Determined priced items to their carts when punching out to Oracle Exchange. Oracle Exchange allows suppliers to specify To Be Determined (TBD) prices for items; however, items without prices are not permitted in Oracle iProcurement. Oracle Exchange automatically disallows adding TBD items to Oracle iProcurement requesters' shopping carts.

6 Requester completes checkout; Oracle iProcurement processes order

Oracle iProcurement adds the items to the requisition. It also references Oracle e-Commerce Gateway to see if code conversion (mapping) is set up; if it is, Oracle e-Commerce Gateway performs the required conversions between the external code values coming from the supplier and the internal code values set up in Oracle Applications. If the Oracle iProcurement administrator, in the eContent Manager, set the option to prevent modifications to items for that punchout definition, then the requester cannot make changes to the items returned from the punchout.

Oracle iProcurement converts the currency in which the items are priced to the functional currency used by Oracle iProcurement.

If the category of the item has an information template associated with it, a link displays next to the item in the cart just before checkout. If the information template has required fields, the requester must click the link to access the template before being allowed to check out.

The requisition then goes through the normal workflow, approval, and purchase order processes configured in Oracle iProcurement and Oracle Purchasing.

Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)

The following diagram shows the flow for Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML).

Figure A–2   Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML)
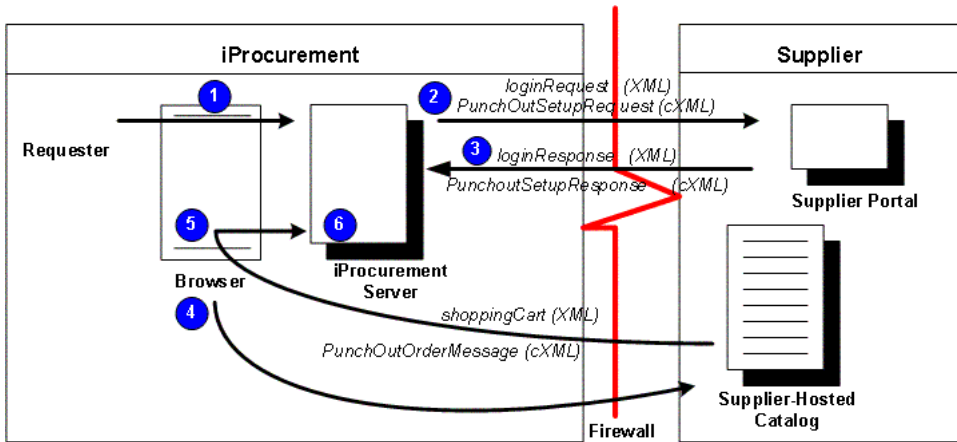
The following sections describe the steps in Models 2a and 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML or cXML) in detail.

1 Requester logs on to Oracle iProcurement

The requester logs on to Oracle iProcurement.

2 Requester clicks punchout link to supplier site

From Oracle iProcurement, the requester clicks a punchout link to the supplier site. When the requester clicks the link, the browser connects to the Oracle iProcurement server.

If punching out to a secure supplier site, the Oracle iProcurement server first calls Oracle secure socket layer (SSL) application programmable interfaces (APIs) to establish a secure connection with the site. The APIs request the Certification Authority (CA) digital certificate from the secure supplier site. This digital certificate is compared to certificate authorities stored in the ca-bundle.crt file in Oracle iProcurement.

After the connection is established, Oracle iProcurement generates the *loginRequest* XML document, which includes a base set of user details and the return URL for the Oracle iProcurement instance. (This XML document can contain additional user details if the option in the eContent manager punchout setup pages in Oracle iProcurement was selected to send additional details.) If the supplier uses cXML,

Oracle iProcurement sends the *PunchOutSetupRequest* cXML document to the supplier site. Oracle iProcurement passes the request document to the supplier site in an HTTP request using the POST format.

3 Supplier authenticates requester and returns response

The supplier receives the loginRequest document (or PunchOutSetupRequest cXML document) and authenticates the Oracle iProcurement requester. (When the supplier sets up access to its site as described in this guide, this logon is invisible to the requester; the requester does not physically log on again to the supplier site.) Next, the supplier must generate the *loginResponse* XML document accepting the requester's logon. If the supplier uses cXML, the supplier generates a response in the form of a *PunchOutSetupResponse* cXML document. The response must be sent in an HTTP response.

It is recommended that the supplier site be *session-aware* or have some other method for distinguishing logons to its site. The Oracle iProcurement setup for the buyer typically creates one logon user account—a single proxy user name that is used by all requesters in the company to access the supplier site. In this case, the supplier should use a session key to assign to each login made by that account, using the URL in the response document, or develop some other method to allow multiple users with the same login account to access its site at one time. For example, the supplier can use some other unique identifier sent with the user's information, in addition to the single login account, to identify the user.

4 Oracle iProcurement redirects browser to supplier site for shopping

Now that a secure and trusted connection is established, Oracle iProcurement redirects the requester's browser to the supplier site—to the login URL provided by the supplier. Now the requester is allowed to search for and add items to the shopping cart on the supplier site.

5 Requester finishes shopping; supplier site returns cart to Oracle iProcurement

When the requester finishes adding items to the cart on the supplier site, the supplier site returns the shopping cart to the Oracle iProcurement server via the requester's browser. The return to the requester's browser is invisible to the requester. Specifically, the browser uses the HTML FORM POST action to pass the items in the shopping cart to Oracle iProcurement. The cart must be contained in a hidden form field. To post the shopping cart items, the FORM POST must use the return URL that Oracle iProcurement sent in the loginRequest document. The

parameter for the post should be oracleCart. (The oracleCart parameter contains the entire shopping cart contents.)

The supplier site must return the shopping cart information in a url-encoded *shoppingCart* XML document. If using cXML, the supplier site returns the shopping cart information to Oracle iProcurement in the *PunchOutOrderMessage* cXML document. See the *cXML User's Guide* at http://www.cxml.org for more information on cXML requirements.

6 Requester completes checkout; Oracle iProcurement processes order

Oracle iProcurement adds the items to the requisition. It also references Oracle e-Commerce Gateway to see if code conversion (mapping) is set up; if it is, Oracle e-Commerce Gateway performs the required conversions between the external code values coming from the supplier and the internal code values set up in Oracle Applications. If the Oracle iProcurement administrator, in the eContent Manager, set the option to prevent modifications to items for that punchout definition, then the requester cannot make changes to the items returned from the punchout.

Oracle iProcurement converts the currency in which the items are priced to the functional currency used by Oracle iProcurement.

If the category of the item has an information template associated with it, a link displays next to the item in the cart just before checkout. If the information template has required fields, the requester must click the link to access the template before being allowed to check out.

The requisition then goes through the normal workflow, approval, and purchase order processes configured in Oracle iProcurement and Oracle Purchasing.

Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML)

The following diagram shows the flow for Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML).
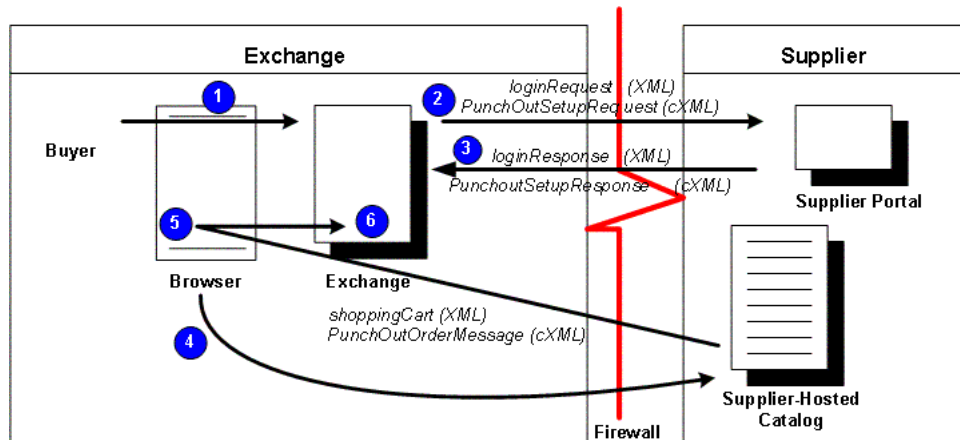
The following sections describe the steps in Models 3a and 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML or cXML) in detail.

1 Buyer logs on to Oracle Exchange

The buyer logs on to Oracle Exchange.

2 Buyer clicks punchout link to supplier site

On Oracle Exchange, the buyer clicks a punchout link directly to the supplier site from the **Shopping** home page or **Search Results** page. When the buyer clicks the link, the browser connects to the Oracle Exchange server, which establishes a connection to the supplier site.

If the supplier site is a secure site, Oracle Exchange first calls Oracle secure socket layer (SSL) application programmable interfaces (APIs) to establish a secure connection with the site. The APIs request the Certification Authority (CA) digital certificate from the site. This digital certificate is compared to certificate authorities stored in the pomdigcrt.txt file in Oracle Exchange.

After the connection is established, Oracle Exchange generates the *loginRequest* XML document, which includes a base set of user details and the return URL for Oracle Exchange. Oracle Exchange passes this XML document to the supplier site. If the supplier uses cXML, Oracle Exchange passes the *PunchOutSetupRequest* cXML document to the supplier site.

3 Supplier authenticates buyer

The supplier receives the loginRequest XML document (or PunchOutSetupRequest cXML document) and authenticates the buyer. (When the supplier sets up access to its site as described in this guide, this logon is invisible to the buyer; the buyer does not physically log on again to the supplier site.) Next, the supplier must generate the *loginResponse* XML document and send the document back to Oracle Exchange to accept the buyer's logon. If the supplier uses cXML, the supplier generates a response in the form of a *PunchOutSetupResponse* cXML document. The response must be sent in an HTTP response.

4 Oracle Exchange redirects buyer's browser to supplier site for shopping

Now that a secure and trusted connection is established, Oracle Exchange redirects the requester's browser to the supplier site—to the login URL provided by the supplier. Now the requester is allowed to search for and add items to the shopping cart on the supplier site.

5 Buyer finishes shopping; supplier site returns cart to Oracle Exchange

When the buyer finishes adding items to the cart on the supplier's site, the supplier site returns the shopping cart to the Oracle Exchange server via the buyer's browser. The return to the buyer's browser is invisible to the buyer. Specifically, the browser uses the HTML FORM POST action to pass the items in the shopping cart to the Oracle Exchange server. The cart must be contained in a hidden form field. To post the shopping cart items, the FORM POST must use the return URL that Oracle Exchange sent in the loginRequest document. The parameter for the post should be oracleCart. (The oracleCart parameter contains the entire shopping cart contents.)

The supplier site must return the shopping cart information in a url-encoded *shoppingCart* XML document. If using cXML, the supplier site returns the shopping cart information to Oracle Exchange in the *PunchOutOrderMessage* cXML document. See the *cXML User's Guide* at http://www.cxml.org for more information on cXML requirements.

6 Buyer completes checkout process; Oracle Exchange processes order

Oracle Exchange adds the items to the order. If the buyer and supplier companies have set up data mapping on Oracle Exchange, Exchange performs data mapping at this step, when the cart is returned to the Oracle Exchange server for processing.

If the items are priced in different currencies, Oracle Exchange creates one order for each currency.

The buyer cannot make changes to the punchout items on the order.

Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

The following diagram shows the flow for Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML).

Figure A–4 Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)
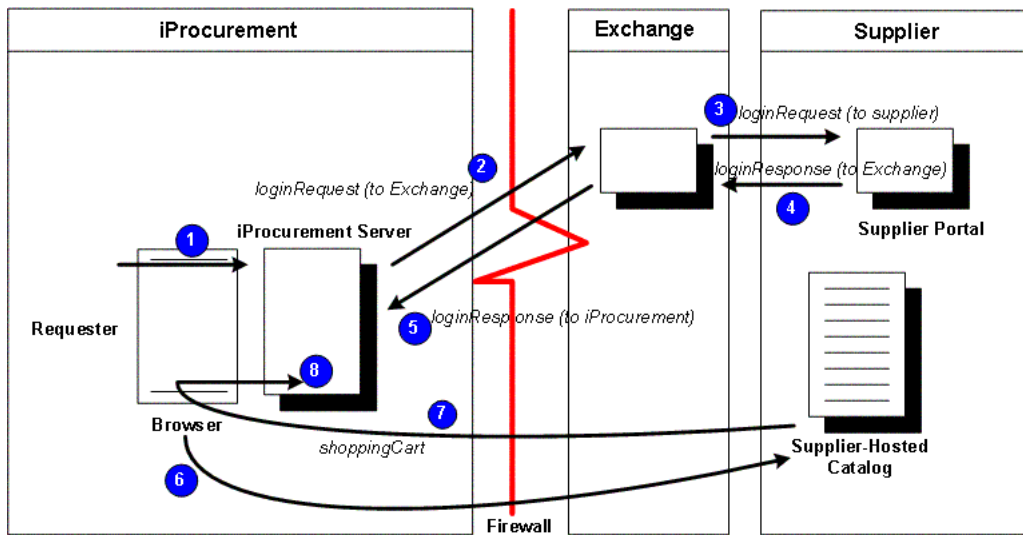
The following sections describe the steps in Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML) in detail.

1 Requester logs on to Oracle iProcurement

The requester logs on to Oracle iProcurement.

2 Requester clicks punchout link to supplier site

From Oracle iProcurement, the requester clicks a punchout link to the supplier site. The link goes through Oracle Exchange. When the requester clicks the link, the browser connects to the Oracle iProcurement server, which establishes a secure

connection to Oracle Exchange (if the Exchange is secure) and requests the digital certificate from Oracle Exchange. (Exchange.Oracle.com is a secure site.)

The Oracle iProcurement server first calls Oracle secure socket layer (SSL) application programmable interfaces (APIs) to establish a secure connection with Oracle Exchange. The APIs request the Certification Authority (CA) digital certificate from Oracle Exchange. This digital certificate is compared to certificate authorities stored in the ca-bundle.crt file in Oracle iProcurement.

After the secure connection is established, Oracle iProcurement generates the *loginRequest* XML document, which includes a base set of user details and the return URL for the Oracle iProcurement instance. Oracle iProcurement passes this XML document to the Oracle Exchange site in an HTTP request using the POST format. When the punchout is set up as described in this guide, this logon is invisible to the requester; the requester does not physically log on again to Oracle Exchange.

3 Oracle Exchange authenticates requester

Oracle Exchange receives the loginRequest and authenticates the Oracle iProcurement user. The user name and password used to authenticate the requester are the same as the proxy user name and password that the Oracle iProcurement administrator created when registering the proxy user on Oracle Exchange.

Oracle Exchange then uses Oracle SSL APIs to establish a secure connection to the supplier site (if the site requires), opens a connection between itself and the supplier site, and generates the *loginRequest* XML document in an HTTP request using the POST format. The document includes the site-level password that the supplier requires to access its Web catalog, the Oracle iProcurement return URL, and the Oracle iProcurement user and company details that were in the initial loginRequest from Oracle iProcurement to Oracle Exchange.

> **Note:** The supplier creates the site-level password on Oracle Exchange. The Oracle iProcurement administrator does not need to store or maintain this information.

It is recommended that the supplier site be *session-aware* or have some other method for distinguishing logons to its site. The Oracle iProcurement setup for the buyer typically creates one logon user account—a single proxy user name that is used by all requesters in the company to access the supplier site. In this case, the supplier should use a session key to assign to each login made by that account, using the <loginURL> tag of the loginResponse document, or develop some other method to

allow multiple users with the same login account to access its site at one time. For example, the supplier can use some other unique identifier sent with the user's information, in addition to the single login account, to identify the user.

4 Supplier site responds to Oracle Exchange

The supplier site generates a response in the form of a *loginResponse* XML document that contains the login URL with session information for the supplier's site.

The supplier site returns the loginResponse document to Oracle Exchange.

5 Oracle Exchange forwards supplier site response to Oracle iProcurement

Oracle Exchange then sends a *loginResponse* document to Oracle iProcurement. The LoginResponse XML document contains the URL that the Oracle iProcurement requester will use to connect to the supplier site.

6 Oracle iProcurement redirects browser to supplier site for shopping

Using the login URL provided in the loginResponse XML document, Oracle iProcurement redirects the Oracle iProcurement requester's browser to the supplier site using the supplier's login URL. The requester is allowed to search for and add items to the shopping cart on the supplier site.

7 Requester finishes shopping; supplier site returns cart to Oracle iProcurement

When the requester finishes adding items to the cart on the supplier site, the supplier site returns the shopping cart to the Oracle iProcurement server via the requester's browser. The return to the requester's browser is invisible to the requester. Specifically, the browser uses the HTML FORM POST action to pass the items in the shopping cart to Oracle iProcurement. The cart must be contained in a hidden form field. To post the shopping cart items, the FORM POST must use the return URL that Oracle iProcurement sent in the loginRequest document. The parameter for the post should be oracleCart. (The oracleCart parameter contains the entire shopping cart contents.)

The supplier site returns the shopping cart information to Oracle iProcurement in the *shoppingCart* XML document. This document must be url-encoded.

8 Requester completes checkout; Oracle iProcurement processes order

Oracle iProcurement adds the items to the requisition. It also references Oracle e-Commerce Gateway to see if code conversion (mapping) is set up; if it is, Oracle e-Commerce Gateway performs the required conversions between the external code values coming from the supplier and the internal code values set up in Oracle Applications. (This model does not use Oracle Exchange data mapping.) If the Oracle iProcurement administrator, in the eContent Manager, set the option to prevent modifications to items for that punchout definition, then the requester cannot make changes to the items returned from the punchout.

Oracle iProcurement converts the currency in which the items are priced to the functional currency used by Oracle iProcurement.

If the category of the item has an information template associated with it, a link displays next to the item in the cart just before checkout. If the information template has required fields, the requester must click the link to access the template before being allowed to check out.

The requisition then goes through the normal workflow, approval, and purchase order processes configured in Oracle iProcurement and Oracle Purchasing.

Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

The following diagram shows the flow for Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML).

Figure A–5   Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)
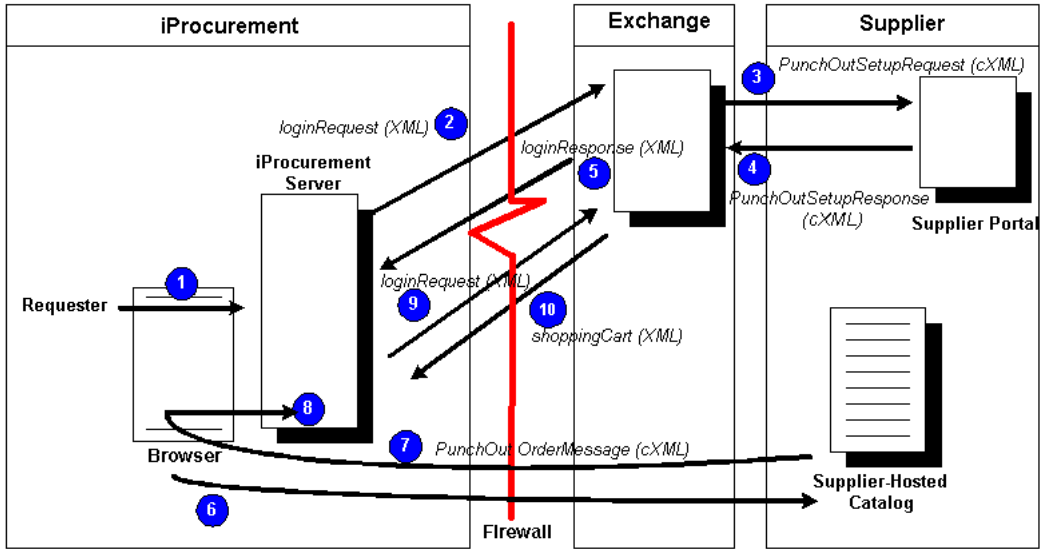
The following sections describe the steps in Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML) in detail.

1 Requester logs on to Oracle iProcurement

The requester logs on to Oracle iProcurement.

2 Requester clicks punchout link to supplier site

From Oracle iProcurement, the requester clicks a punchout link to the supplier site. The punchout goes through Oracle Exchange. When the requester clicks the link, the browser connects to the Oracle iProcurement server, which establishes a secure connection to Oracle Exchange (if the Exchange is secure) and requests the digital certificate from Oracle Exchange. (Exchange.Oracle.com is a secure site.)

The Oracle iProcurement server first calls Oracle secure socket layer (SSL) application programmable interfaces (APIs) to establish a secure connection with Oracle Exchange. The APIs request the Certification Authority (CA) digital

certificate from Oracle Exchange. This digital certificate is compared to certificate authorities stored in the ca-bundle.crt file in Oracle iProcurement.

After the secure connection is established, Oracle iProcurement generates the *loginRequest* XML document, which includes a base set of user details and the return URL for the Oracle iProcurement instance. Oracle iProcurement passes this XML document to the Oracle Exchange site in an HTTP request using the POST format. When the punchout is set up as described in this guide, this logon is invisible to the requester; the requester does not physically log on again to Oracle Exchange.

3 Oracle Exchange authenticates requester

Oracle Exchange receives the loginRequest and authenticates the Oracle iProcurement requester. The user name and password used to authenticate the requester are the same as the proxy user name and password that the Oracle iProcurement administrator created when registering the proxy user on Oracle Exchange.

Oracle Exchange then uses Oracle SSL APIs to establish a secure connection to the supplier site (if the site requires), opens a connection between itself and the supplier site, and generates the *PunchOutSetupRequest* cXML document. The document includes the site-level password that the supplier requires to access its Web catalog, the Oracle iProcurement return URL, and the Oracle iProcurement user and company details that were in the initial loginRequest from Oracle iProcurement to Oracle Exchange.

> **Note:**   The supplier creates the site-level password on Oracle Exchange. The Oracle iProcurement administrator does not need to store or maintain this information.

It is recommended that the supplier site be *session-aware* or have some other method for distinguishing logons to its site. The Oracle iProcurement setup for the buyer typically creates one logon user account—a single proxy user name that is used by all requesters in the company to access the supplier site. In this case, the supplier should use a session key to assign to each login made by that account, using the URL in the punchout response document, or develop some other method to allow multiple users with the same login account to access its site at one time. For example, the supplier can use some other unique identifier sent with the user's information, in addition to the single login account, to identify the user.

4 Supplier site responds to Oracle Exchange

The supplier site then generates a response in the form of a *PunchOutSetupResponse* cXML document, which contains the login URL with session information for the supplier's site. The supplier site returns the PunchOutSetupResponse document to Oracle Exchange.

5 Oracle Exchange forwards supplier site response to Oracle iProcurement

Oracle Exchange then uses the PunchOutSetupResponse to generate the *loginResponse* XML document, and sends the loginResponse document to Oracle iProcurement. The loginResponse XML document contains the URL that the Oracle iProcurement requester will use to connect to the supplier site.

6 Oracle iProcurement redirects requester's browser to supplier site for shopping

Using the login URL provided in the loginResponse XML document, Oracle iProcurement redirects the Oracle iProcurement requester's browser to the supplier site using the supplier's login URL. The requester is allowed to search for and add items to the shopping cart on the supplier site.

7 Requester finishes shopping; supplier site returns cart to Oracle iProcurement

When the requester finishes adding items to the cart on the supplier site, the supplier site returns the shopping cart to the Oracle iProcurement server via the requester's browser. The return to the requester's browser is invisible to the requester. Specifically, the browser uses the HTML FORM POST action to pass the items in the shopping cart to Oracle iProcurement, using the return URL that Oracle iProcurement sent during the initial punchout. The cart must be contained in a hidden form field. The parameter for the post should be cxml-urlencoded. (The cxml-urlencoded parameter contains the entire shopping cart contents.)

The supplier site returns the shopping cart information to Oracle iProcurement in the *PunchOutOrderMessage* cXML document. See the *cXML User's Guide* at http://www.cxml.org for more information on cXML requirements.

8 Shopping cart goes to Oracle iProcurement

The cxml-urlencoded parameter used to post the shopping cart lets Oracle iProcurement know this is a cXML shopping cart.

9 Oracle iProcurement redirects cart to Oracle Exchange for conversion

To convert the PunchOutOrderMessage cXML document to XML, Oracle iProcurement receives and redirects the shopping cart to Oracle Exchange. Oracle iProcurement establishes a secure and trusted connection with Oracle Exchange, then forwards the cXML shopping cart to Oracle Exchange via the *loginRequest* document.

10 Oracle Exchange converts the cart to XML and returns it to Oracle iProcurement

Oracle Exchange then converts the shopping cart from cXML to XML and does the code conversions (UOM, currency, and trading partner data mapping), if any, defined on Oracle Exchange.

Oracle Exchange then returns the shopping cart in the form of the *shoppingCart* XML document to Oracle iProcurement for processing, approval routing, and purchase order creation.

If the Oracle iProcurement administrator has set up data mapping in Oracle e-Commerce Gateway, e-Commerce Gateway performs data mapping at this step, when the cart is returned to the Oracle iProcurement server for processing.

If the category of the item has an information template associated with it, a link displays next to the item in the cart just before checkout. If the information template has required fields, the requester must click the link to access the template before being allowed to check out.

Oracle iProcurement converts the currency in which the items are priced to the functional currency used by Oracle iProcurement.

The requisition then goes through the normal workflow, approval, and purchase order processes configured in Oracle iProcurement and Oracle Purchasing.

Models 1 and 2: Transparent Punchout to Oracle Exchange or Supplier (XML)

The following diagram shows the flow for Models 1 and 2: Transparent Punchout to Oracle Exchange or Supplier (XML).

Figure A–6   Models 1 and 2: Transparent Punchout to Oracle Exchange or Supplier (XML)
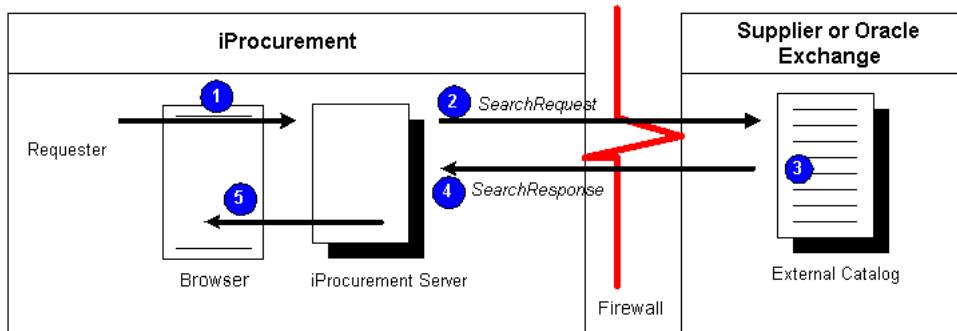
The following sections describe the steps in Models 1 and 2: Transparent Punchout to Oracle Exchange or Supplier (XML) in detail.

1 Requester conducts search in Oracle iProcurement

If you have configured stores in the eContent Manager, the requester selects a store before searching. Otherwise, the requester enters a search term into the default or favorite store.
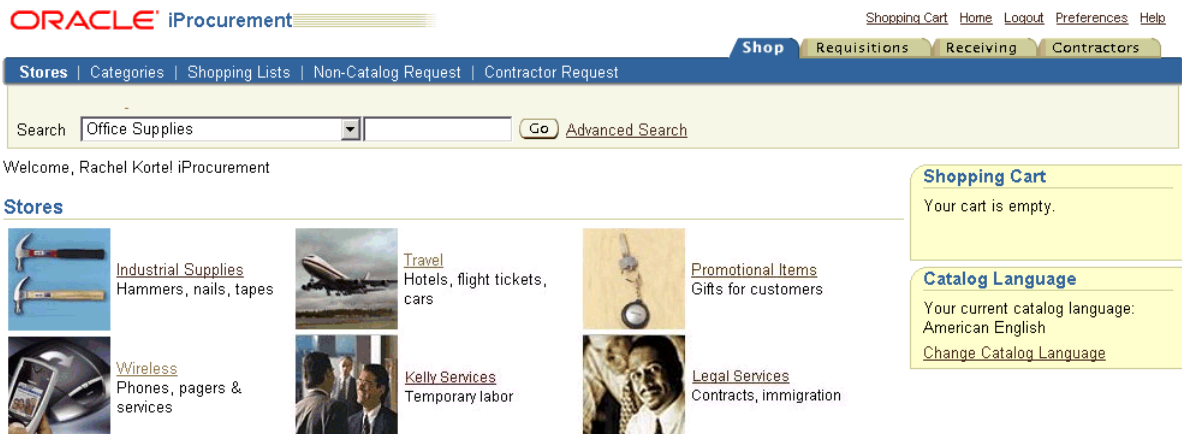
2 Oracle iProcurement sends search request XML document to site

If the store includes a transparent punchout catalog, Oracle iProcurement sends a search request XML document to the external site.

First, the browser connects to the Oracle iProcurement server. The Oracle iProcurement server then calls Oracle secure socket layer (SSL) application programmable interfaces (APIs) to establish a secure connection with the site, if the site is secure. (Exchange.Oracle.com is a secure site.) The APIs request the Certification Authority (CA) digital certificate from the site. This digital certificate is compared to certificate authorities stored in the ca-bundle.crt file in Oracle iProcurement. (See Appendix C for more information on how secure connections work.)

After the secure connection is established, the Oracle iProcurement server generates the *ItemSearchRequest* XML document, which includes a base set of user details for the Oracle iProcurement instance. Oracle iProcurement passes this XML document to the site via an HTTP request using the POST format.

3 Site processes request and generates search results

The external site receives the ItemSearchRequest document, authenticates the Oracle iProcurement requester, and generates search results.

The supplier can optionally return an authenticated key in the ItemSearchResponse document using the <AuthenticatedKey> field. When Oracle iProcurement receives this key, it includes the key in every subsequent search request to this supplier. The supplier site can validate the key against the key it previously sent and therefore does not need to authenticate the user again. Use of the authenticated key improves the performance of the transparent punchout.

Suppliers with Web stores typically also have a search engine for that store. The supplier site should construct a query from the search criteria in the ItemSearchRequest document and use the search engine to execute the search request. Alternatively, the site can execute the search request on the catalog directly (as long as the search results are returned in the ItemSearchResponse document).

4 Site returns search results XML document to Oracle iProcurement

The site returns the search results XML document to the Oracle iProcurement server in the HTTP response. The supplier site must return the search results information in the *ItemSearchResponse* XML document. See Appendix B for a detailed description of the ItemSearchResponse XML document.

> **Note:** To optimize search performance, transparent punchout does not display the Functional Currency or Functional Currency Price descriptors on the **Search Results** page even if you set them up this way using schema editing.
>
> In a transparent punchout to Oracle Exchange, Oracle iProcurement does not honor price breaks the supplier may have created on Oracle Exchange.

5 Oracle iProcurement displays search results

Oracle iProcurement uses the search results XML document from the external site to display search results in Oracle iProcurement.

Oracle iProcurement displays a set number of search results per page. This number, also known as the Shopping Search Results Per Page, can be set by individual requesters in their preferences. (By default, the Shopping Search Results Per Page is 7.) Transparent punchout returns two pages of results at a time. For example, if the Shopping Search Results Per Page is 7, transparent punchout returns two pages of search results, each page with seven items.

When the requester clicks the "Next" link on the second page (and every second page thereafter), the system sends a new ItemSearchRequest document to fetch the rest of the results, two pages at a time. This ItemSearchRequest document is identical to the first except the <startResult> value changes. For details, see ItemSearchRequest on page B-43.

Likewise, if the requester sorts the transparent punchout results (you can sort transparent punchout results by Price only), a new ItemSearchRequest document is sent. This ItemSearchRequest document is identical to the first except that it specifies a <sortBy> value. If the supplier can fulfill the sort request, sorted search results are displayed. Otherwise, the supplier can use an ItemSearchResponse error code in the <Status> field to indicate that it was unable to sort the results.

> **Note:** Oracle iProcurement cannot enforce how the supplier interprets the search. For example, Oracle iProcurement cannot verify if the results are sorted correctly.

After viewing the transparent punchout results, the requester clicks Add to Cart for the desired items, and Oracle iProcurement adds the items to the requisition. Oracle iProcurement also references Oracle e-Commerce Gateway to see if code conversion (mapping) is set up; if it is, Oracle e-Commerce Gateway performs the required conversions between the external code values coming from the supplier and the internal code values set up in Oracle Applications.

> **Note:** Although a transparent punchout to Oracle Exchange returns To Be Determined (TBD) priced items to Oracle iProcurement, requesters cannot add TBD items to their shopping cart. (Oracle Exchange allows suppliers to specify TBD prices for items; however, items without prices are not permitted in Oracle iProcurement.) When the requester attempts to add a TBD-priced item to the shopping cart in Oracle iProcurement, an error message informs the requester that TBD items cannot be added to the cart. The buyer should be aware that although TBD items present no problems for transparent punchout, the TBD items will be rejected by the shopping cart in Oracle iProcurement. Buyers and suppliers should resolve this together, if required. The supplier could remove items with TBD prices, or the buyer could use the pricing approval feature on Oracle Exchange to reject these prices.

Oracle iProcurement converts the currency in which the items are priced to the functional currency used by Oracle iProcurement.

The requester then completes the checkout process for the items in the shopping cart, and Oracle iProcurement processes the order. The requisition goes through the normal workflow, approval, and purchase order processes configured in Oracle iProcurement and Oracle Purchasing.

B

DTDs, Documents, and Descriptions

This chapter covers the following topics:

DTDs and Documents by Model

All the document type definition (DTD) documents used by XML punchout and transparent punchout are displayed in this appendix. For cXML punchout documents, the DTD can be downloaded from the following URL: http://www.cxml.org.

The following tables list the punchout documents and DTDs used for each model.

For a graphical description of the punchout and transparent punchout documents by model, see the illustrations in Appendix A.

Table B–1   Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML) Documents

Reference	DTD
loginRequest on page B-9	OraclePunchout.dtd
loginResponse on page B-22	OraclePunchout.dtd
shoppingCart on page B-23	OraclePunchout.dtd

Table B–2   Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML) Documents

Reference	DTD
loginRequest on page B-9	OraclePunchout.dtd
loginResponse on page B-22	OraclePunchout.dtd
shoppingCart on page B-23	OraclePunchout.dtd

Table B–3   Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML) Documents

Reference	DTD
PunchOutSetupRequest on page B-32	cXML.dtd
PunchOutSetupResponse on page B-39	cXML.dtd
PunchOutOrderMessage on page B-40	cXML.dtd

Table B–4   Model 3a: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML) Documents

Reference	DTD
loginRequest on page B-9	OraclePunchout.dtd
loginResponse on page B-22	OraclePunchout.dtd
shoppingCart on page B-23	OraclePunchout.dtd

Table B–5   Model 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (cXML) Documents

Reference	DTD
PunchOutSetupRequest on page B-32	cXML.dtd
PunchOutSetupResponse on page B-39	cXML.dtd
PunchOutOrderMessage on page B-40	cXML.dtd

Table B–6   Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML) Documents

Reference	DTD
loginRequest on page B-9	OraclePunchout.dtd
loginResponse on page B-22	OraclePunchout.dtd
shoppingCart on page B-23	OraclePunchout.dtd
supplierSync on page B-31	OraclePunchout.dtd

Table B–7   Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML) Documents

Reference	DTD
loginRequest on page B-9	OraclePunchout.dtd
PunchOutSetupRequest on page B-32	cXML.dtd
PunchOutSetupResponse on page B-39	cXML.dtd
loginResponse on page B-22	OraclePunchout.dtd
PunchOutOrderMessage on page B-40	cXML.dtd
shoppingCart on page B-23	OraclePunchout.dtd
supplierSync on page B-31	OraclePunchout.dtd

Table B–8   Models 1 and 2: Transparent Punchout to Oracle Exchange or Supplier (XML)

Reference	DTD
ItemSearchRequest on page B-43	ItemSearchRequest.dtd
ItemSearchResponse on page B-50	ItemSearchResponse.dtd

DTDs, Documents, and Descriptions

The following sections provide the DTDs, examples, and descriptions of each punchout and transparent punchout document.

The Required or Optional indicators below refer to whether a value in the field is required.

Note: Also note the DTD structure. For example, a bar (|) between values indicates that you should provide only one of the values.

OraclePunchout.dtd

The OraclePunchout.dtd covers all XML punchout documents used by all XML punchout models:

- loginRequest

- loginResponse

- shoppingCart

- supplierSync

The OraclePunchout.dtd is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Version 1.0 of Oracle Punchout DTD
     User by Oracle Exchange 6.2.4 and Oracle iProcurement Release FPI
 -->

<!-- root elements -->
<!ELEMENT request (header,body)>
<!ELEMENT response (header,body)>

<!-- different possible bodies -->
<!-- loginRequest (request) -->
<!-- loginResponse (response) -->
<!-- shoppingCart (response) -->
<!-- supplierSync (response) -->
<!ELEMENT body (
(loginInfo,searchKeywords?,icxSessionCallBackURL?,partySiteId?,cxmlCart?)|
loginURL|
OrderLinesDataElements|
supplier*
```

```
)>

<!--  START header elements -->
<!ELEMENT header
(requestID?,login?,(action|return),cookies?,language?,searchLanguage?,userArea?)
>
  <!-- version to identify the DTD used, current version is 1.0 -->
  <!ATTLIST header version CDATA #REQUIRED>
  <!-- identifier in an asynchronous request/response communication, for future
use -->
  <!ELEMENT requestID (#PCDATA)>
  <!-- login information. session is for future use -->
  <!ELEMENT login ((username,password)|session)>
    <!ELEMENT username (#PCDATA)>
    <!ELEMENT password (#PCDATA)>
    <!ELEMENT session (sessionID)>
    <!ELEMENT sessionID (#PCDATA)>
  <!-- for request only, to indicate the type of the request -->
  <!ELEMENT action (#PCDATA)>
  <!-- for response only -->
  <!ELEMENT return (returnMessage?)>
    <!-- return code for the request. S = Success, E = Error, U = Unexpected,
        W = Warning and A = Authentication Failure. Exchange and IP only
        uses S, E, and A
    -->
    <!ATTLIST return returnCode (S|E|U|W|A) #REQUIRED>
    <!ELEMENT returnMessage (#PCDATA)>
  <!-- cookies to identify the session, for future use -->
  <!ELEMENT cookies (cookie*)>
    <!ELEMENT cookie (#PCDATA)>
    <!ATTLIST cookie name CDATA #REQUIRED>
  <!-- user's preferred session language only -->
  <!ELEMENT language (#PCDATA)>
  <!-- user's preferred search language only -->
  <!ELEMENT searchLanguage (#PCDATA)>
  <!-- any additional customizable info -->
  <!ELEMENT userArea ANY>
<!-- END header elements -->

<!-- START additional elements in userArea which may be populated by IP,
included for DTD validation -->
<!ELEMENT operatingUnit (#PCDATA)>
<!ELEMENT shipTo (#PCDATA)>
<!ELEMENT deliverTo (#PCDATA)>
<!ELEMENT fullName (#PCDATA)>
```

```
<!ELEMENT workPhone (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT manager (#PCDATA)>
<!ELEMENT managerEmail (#PCDATA)>
<!ELEMENT managerPhone (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT dateFormat (#PCDATA)>
<!-- END additional elements in userArea -->

<!-- START login request elements -->
<!ELEMENT loginInfo (exchangeInfo?,userInfo,returnURL)>
  <!-- exchange name if the request is from exchange -->
  <!ELEMENT exchangeInfo (exchangeName)>
    <!ELEMENT exchangeName (#PCDATA)>
  <!ELEMENT userInfo (userName,appUserName,userContactInfo?,userCompany?)>
    <!-- full name of user -->
    <!ELEMENT userName (#PCDATA)>
    <!-- username of user in the application -->
    <!ELEMENT appUserName (#PCDATA)>
    <!ELEMENT userContactInfo (userPhone?,userEmail?)>
      <!ELEMENT userPhone (#PCDATA)>
      <!ELEMENT userEmail (#PCDATA)>
    <!ELEMENT userCompany (companyName,companyDUNS,contactName?,contactPhone?)>
      <!-- name of the user's company -->
      <!ELEMENT companyName (#PCDATA)>
      <!-- DUNS of the user's company -->
      <!ELEMENT companyDUNS (#PCDATA)>
      <!ELEMENT contactName (#PCDATA)>
      <!ELEMENT contactPhone (#PCDATA)>
  <!-- URL the shopping cart will be returned to -->
  <!ELEMENT returnURL (#PCDATA)>
<!-- search text entered by the users -->
<!ELEMENT searchKeywords (#PCDATA)>
<!-- URL which can be pinged constantly to keep IP session alive -->
<!ELEMENT icxSessionCallBackURL (#PCDATA)>
<!-- party id of the supplier to punchout to via OEX -->
<!ELEMENT partySiteId (#PCDATA)>
<!-- CXML shopping cart sent from IP to OEX for mapping -->
<!ELEMENT cxmlCart ANY>
<!-- END login request elements -->

<!-- START login response elements -->
<!ELEMENT loginURL (#PCDATA)>
<!-- END login response elements -->
```

```
<!-- START shopping cart elements -->
<!ELEMENT OrderLinesDataElements (catalogTradingPartner?, orderLine*)>
  <!ELEMENT catalogTradingPartner (#PCDATA)>
  <!ELEMENT orderLine (contract?, item, category, price, supplier,
additionalAttributes?)>

<!-- DTD for element CONTRACT -->
<!ELEMENT contract
((supplierContract|buyerContract),buyerContractLineNumber?,catalogType?)>
  <!ATTLIST contract contractNumberIdentifier (KNOWN|UNKNOWN|INFORMATIONAL|NONE)
#IMPLIED>
  <!ELEMENT supplierContract (contractNumber)>
  <!ELEMENT buyerContract (contractNumber)>
  <!ELEMENT contractNumber (#PCDATA)>
  <!ELEMENT buyerContractLineNumber (#PCDATA)>
  <!ELEMENT catalogType (#PCDATA)>

<!-- DTD for element ITEM -->
<!ELEMENT item (itemNumber, itemDescription, unitOfMeasure, hazardClass?)>
  <!ATTLIST item lineType (GOODS|SERVICES_AMOUNT|SERVICES_QUANTITY) #IMPLIED>
  <!ATTLIST item quantity CDATA "1">
  <!ELEMENT itemNumber ((supplierItemNumber|buyerItemNumber),
manufacturerItemNumber?)>
  <!ELEMENT supplierItemNumber (itemID, supplierReferenceNumber?)>
  <!ELEMENT itemID (#PCDATA)>
  <!ELEMENT supplierReferenceNumber (#PCDATA)>
  <!ELEMENT manufacturerItemNumber (itemID, manufacturerName)>
  <!ELEMENT manufacturerName (#PCDATA)>
  <!ELEMENT buyerItemNumber (itemID, buyerItemRevision?)>
  <!ELEMENT buyerItemRevision (#PCDATA)>
  <!ELEMENT itemDescription (#PCDATA)>
  <!ELEMENT unitOfMeasure (buyerUnitOfMeasure|supplierUnitOfMeasure)>
  <!ELEMENT buyerUnitOfMeasure (#PCDATA)>
  <!ELEMENT supplierUnitOfMeasure (supplierUOMType,supplierUOMQuantity?)>
  <!ELEMENT supplierUOMType (#PCDATA)>
  <!ELEMENT supplierUOMQuantity (#PCDATA)>
  <!ELEMENT hazardClass (#PCDATA)>

<!-- DTD for element CATEGORY -->
<!ELEMENT category (categoryCode)>
  <!ELEMENT categoryCode (#PCDATA)>
  <!ATTLIST categoryCode categoryCodeIdentifier(SPSC|SUPPLIER|BUYER) #IMPLIED>

<!-- DTD for element PRICE -->
<!ELEMENT price (currency, unitPrice)>
```

```
  <!ELEMENT currency (#PCDATA)>
  <!ELEMENT unitPrice (#PCDATA)>

<!-- DTD for element SUPPLIER -->
<!-- used in shopping cart -->
<!-- used in supplier sync response -->
<!ELEMENT supplier (
((supplierDUNS|supplierTradingPartnerCode), supplierName, supplierSite?,
contactName?, contactPhone?)|
(supplierPartyId, supplierImageUrl, supplierLanguageSpecificInfo*)
)>
<!ELEMENT supplierDUNS (#PCDATA)>
<!ELEMENT supplierTradingPartnerCode (#PCDATA)>
<!ELEMENT supplierName (#PCDATA)>
<!ELEMENT supplierSite (#PCDATA)>

<!-- DTD for element ADDITIONAL ATTRIBUTES -->
<!ELEMENT additionalAttributes (attribute1?, attribute2?, attribute3?,
attribute4?, attribute5?, attribute6?, attribute7?, attribute8?, attribute9?,
attribute10?, attribute11?, attribute12?, attribute13?, attribute14?,
attribute15?)>
  <!ELEMENT attribute1 (#PCDATA)>
  <!ELEMENT attribute2 (#PCDATA)>
  <!ELEMENT attribute3 (#PCDATA)>
  <!ELEMENT attribute4 (#PCDATA)>
  <!ELEMENT attribute5 (#PCDATA)>
  <!ELEMENT attribute6 (#PCDATA)>
  <!ELEMENT attribute7 (#PCDATA)>
  <!ELEMENT attribute8 (#PCDATA)>
  <!ELEMENT attribute9 (#PCDATA)>
  <!ELEMENT attribute10 (#PCDATA)>
  <!ELEMENT attribute11 (#PCDATA)>
  <!ELEMENT attribute12 (#PCDATA)>
  <!ELEMENT attribute13 (#PCDATA)>
  <!ELEMENT attribute14 (#PCDATA)>
  <!ELEMENT attribute15 (#PCDATA)>

<!-- END shopping cart elements -->

<!-- START supplier sync elements -->
<!ELEMENT supplierLanguageSpecificInfo
(language,supplierName,supplierDescription,supplierKeywords)>
<!ELEMENT supplierDescription (#PCDATA)>
<!ELEMENT supplierKeywords (#PCDATA)>
<!ELEMENT supplierPartyId (#PCDATA)>
```

```
<!ELEMENT supplierImageUrl (#PCDATA)>
<!-- END supplier sync elements -->
```

loginRequest

Oracle iProcurement generates a loginRequest after establishing a connection with Oracle Exchange in the following models:

- Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML).

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML).

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML).

The loginRequest document is also used as the *supplier sync request* when downloading punchouts from Oracle Exchange.

Oracle iProcurement or Oracle Exchange sends a loginRequest to the supplier in the following models:

- Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML).

- Model 3a: Punchout from Oracle Exchange to Supplier-Hosted Catalog (XML)

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

Example loginRequest to Oracle Exchange

The following is an example loginRequest document used by Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML):

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<request>
   <header version="1.0">
      <login>
         <username>
            <![CDATA[bcbuyeradmin1]]>
         </username>
         <password>
            <![CDATA[welcome]]>
         </password>
      </login>
      <action>
```

```
                    <![CDATA[shopping]]>
                </action>
                <language>
                    <![CDATA[US]]>
                </language>
                <searchLanguage/>
                <userArea>
                    <operatingUnit>
                        <![CDATA[204]]>
                    </operatingUnit>
                    <shipTo/>
                    <deliverTo/>
                </userArea>
            </header>
            <body>
                <loginInfo>
                    <exchangeInfo>
                        <exchangeName>
                            <![CDATA[Oracle iProcurement]]>
                        </exchangeName>
                    </exchangeInfo>
                    <userInfo>
                        <userName>
                            <![CDATA[Stock, Pat]]>
                        </userName>
                        <userContactInfo>
                            <userPhone/>
                            <userEmail>
                                <![CDATA[pat.stock@vision.com]]>
                            </userEmail>
                        </userContactInfo>
                        <appUserName>
                            <![CDATA[OPERATIONS]]>
                        </appUserName>
                        <userCompany>
                            <companyName>
                                <![CDATA[Buyer Corp.]]>
                            </companyName>
                            <companyDUNS/>
                            <contactName/>
                            <contactPhone/>
                        </userCompany>
                    </userInfo>
                    <returnURL>
                        <![CDATA[http://qapache.us.oracle.com:15671/OA_
```

```
HTML/OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_CALLBACK&OAHP=ICX_POR_HOMEPAGE_
MENU&OASF=ICX_CAT_PUNCHOUT_CALLBACK&transactionid=1577779317]]>
        </returnURL>
      </loginInfo>
      <partySiteId/>
      <searchKeywords/>
      <icxSessionCallBackURL/>
   </body>
</request>
```

Example loginRequest to Supplier

Below are example loginRequest XML documents sent to the supplier.

Without Optional Extended Data The following is an example loginRequest XML
document that does not include optional extended user and company information;
this example is used by Model 2a: Punchout from Oracle iProcurement Directly to
Supplier-Hosted Catalog (XML):

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<request>
   <header version="1.0">
      <login>
         <username/>
         <password>welcome</password>
      </login>
      <action>shopping</action>
      <language>EN</language>
      <searchLanguage/>
      <userArea>
         <operatingUnit>204</operatingUnit>
         <shipTo>V1 - New York City</shipTo>
         <deliverTo>V1 - New York City</deliverTo>
      </userArea>
   </header>
   <body>
      <loginInfo>
         <exchangeInfo>
            <exchangeName>Oracle iProcurement</exchangeName>
         </exchangeInfo>
         <userInfo>
            <userName>Stock, Pat</userName>
            <userContactInfo>
               <userPhone/>
               <userEmail>pat.stock@vision.com</userEmail>
```

```
            </userContactInfo>
            <appUserName>OPERATIONS</appUserName>
            <userCompany>
                <companyName>Buyer Corp.</companyName>
                <companyDUNS>144709193</companyDUNS>
                <contactName/>
                <contactPhone/>
            </userCompany>
        </userInfo>
        <returnURL>http://qapache.us.oracle.com:15671/OA_
HTML/OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_CALLBACK&#38;OAHP=ICX_POR_HOMEPAGE_
MENU&#38;OASF=ICX_CAT_PUNCHOUT_CALLBACK&#38;transactionid=1577779317</returnURL>
        </loginInfo>
        <partySiteId/>
        <searchKeywords/>
        <icxSessionCallBackURL/>
    </body>
</request>
```

With Optional Extended Data The following is an example SupplierLoginRequest XML document that includes optional extended user and company information; this example is used by Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML):

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<request>
    <header version="1.0">
        <login>
            <username/>
            <password>welcome</password>
        </login>
        <action>shopping</action>
        <language>EN</language>
        <searchLanguage/>
        <userArea>
            <operatingUnit>204</operatingUnit>
            <shipTo>V1- New York City</shipTo>
            <deliverTo>V1- New York City</deliverTo>
            <fullName>Stock, Ms. Pat</fullName>
            <title>MS.</title>
            <manager>Brown, Ms. Casey</manager>
            <managerEmail>cbrown@vision.com</managerEmail>
            <location>V1- New York City</location>
            <language>US</language>
            <currency>USD</currency>
```

```
            <dateFormat>DD-MON-RRRR</dateFormat>
        </userArea>
    </header>
    <body>
        <loginInfo>
            <exchangeInfo>
                <exchangeName>Oracle iProcurement</exchangeName>
            </exchangeInfo>
            <userInfo>
                <userName>Stock, Pat</userName>
                <userContactInfo>
                    <userPhone/>
                    <userEmail>pat.stock@vision.com</userEmail>
                </userContactInfo>
                <appUserName>OPERATIONS</appUserName>
                <userCompany>
                    <companyName>Buyer Corp.</companyName>
                    <companyDUNS>144709193</companyDUNS>
                    <contactName/>
                    <contactPhone/>
                </userCompany>
            </userInfo>
            <returnURL>http://qapache.us.oracle.com:15671/OA_
HTML/OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_CALLBACK&#38;OAHP=ICX_POR_HOMEPAGE_
MENU&#38;OASF=ICX_CAT_PUNCHOUT_CALLBACK&#38;transactionid=1577779317</returnURL>
</loginInfo>
        <partySiteId/>
        <searchKeywords/>
        <icxSessionCallBackURL/>
    </body>
</request>
```

Sent by Oracle Exchange The following is an example loginRequest XML document sent by Oracle Exchange to the supplier:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<request>
    <header version="1.0">
        <login>
            <username>_username</username>
            <password>welcome</password>
        </login>
        <action>shopping</action>
        <language>EN</language>
        <searchLanguage/>
```

```
            <userArea>
               <operatingUnit/>
               <shipTo/>
               <deliverTo/>
            </userArea>
        </header>
        <body>
            <loginInfo>
                <exchangeInfo>
                   <exchangeName>Oracle Exchange</exchangeName>
                </exchangeInfo>
                <userInfo>
                    <userName>Elvis Impersonator</userName>
                    <userContactInfo>
                       <userPhone>650 633-8239</userPhone>
                       <userEmail>elvis@oracle.com</userEmail>
                    </userContactInfo>
                    <appUserName>ELVIS</appUserName>
                    <userCompany>
                       <companyName>Elvis Corporation</companyName>
                       <companyDUNS>144709193</companyDUNS>
                       <contactName>ELVIS</contactName>
                       <contactPhone>650 633-8239</contactPhone>
                    </userCompany>
                </userInfo>
<returnURL>https://exchange.oracle.com/orders/PunchoutCallback.jsp</returnURL>
            </loginInfo>
            <partySiteId/>
            <searchKeywords/>
            <icxSessionCallBackURL/>
        </body>
</request>
```

Field Descriptions

The fields in the loginRequest document are described below.

<username> Required in loginRequest to Oracle Exchange

Oracle Exchange proxy user name assigned to the Oracle iProcurement requester. This field is used only in a loginRequest sent to Oracle Exchange.

> **Note:** The username is not used (its value is omitted) in a direct punchout to the supplier. In a direct punchout to the supplier, only the password is used to log in.

<password> Required

In a loginRequest to Oracle Exchange, this is the Oracle Exchange proxy user password assigned to the Oracle iProcurement requester.

In a loginRequest to the supplier, this is the password validated by the supplier site and used by all buyers in the accessing application. The supplier provided this password to the Oracle iProcurement administrator, who set up access to the punchout, or entered this password when defining the punchout on Oracle Exchange.

> **Note:** The password is not encrypted in the XML document. It is up to the transport protocol to encrypt the message to ensure security.

<action> Required

The value is *shopping*, except when downloading a punchout from Oracle Exchange. When downloading a punchout from Oracle Exchange, the action is *suppSync*.

For Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML), the value is *shopping* for the initial login request and *mapping* for the conversion of the cXML cart to XML. (When Oracle iProcurement sends the cart back to Oracle Exchange for conversion, it does so via the loginRequest.)

<language> Required

In a loginRequest to Oracle Exchange, this is the requester's session language in Oracle iProcurement, using the Oracle Applications language code.

In a loginRequest to the supplier, this is the requester's session language in Oracle iProcurement or Oracle Exchange (wherever the punchout is initiated), using the International Standards Organization (ISO) language code.

The session language is the language in which the application is displayed. (In Oracle iProcurement, the session language can be changed by clicking "Preferences" at the top of any page, then selecting a Language. From Oracle Exchange, the

session language can be changed by selecting a language on the **Home** page or the **Edit Personal Information** page in My Profile.)

<searchLanguage> Optional

In a loginRequest to Oracle Exchange, this is the requester's search language in Oracle iProcurement, using the Oracle Applications language code.

In a loginRequest to the supplier, this is the requester's search language in Oracle iProcurement or Oracle Exchange (wherever the punchout is initiated), using the ISO language code.

The search language can be chosen by clicking the "Change Catalog Language" link on the **Shop** home page. (For example, if the session language is English, but catalog items exist only in German, the requester can choose German as the search language. This language link exists only if POR: Change Catalog Language is set to Yes.) This field is present only if the requester or buyer performed a search before accessing the punchout.

<operatingUnit> Optional

Unique identifier for the Oracle iProcurement requester's operating unit. This information is passed on to the supplier in Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML) only.

<shipTo> Optional

Ship-to location name for the Oracle iProcurement requester. This information is passed on to the supplier in Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML) only.

<deliverTo> Optional

Deliver-to location name for the Oracle iProcurement requester. This information is passed on to the supplier in Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML) only.

<fullName> Optional extended data *

Full name of the Oracle iProcurement requester as given in Oracle Applications.

<title> Optional extended data *

Title of the Oracle iProcurement requester as given in Oracle Applications.

<manager> Optional extended data *

Name of the requester's manager as given in Oracle Applications.

<managerEmail> Optional extended data *

Requester's manager's e-mail address as given in Oracle Applications.

<location> Optional extended data *

Deliver-to location name for the Oracle iProcurement requester.

<language> Optional extended data *

Oracle Applications language code.

<currency> Optional extended data *

Requester's company's functional currency.

<dateFormat> Optional extended data *

Requester's date format from the requester's preferences.

<exchangeName> Required

In a punchout from Oracle iProcurement, this is always the value Oracle iProcurement. In a punchout from Oracle Exchange, this is the name of the Exchange.

<userName> Required

For a punchout initiated from Oracle iProcurement, this is the full name of the requester (for example, Green, Mr. Terry). For a punchout initiated from Oracle Exchange, this is the full name of the Exchange buyer.

<userPhone> Optional

Oracle iProcurement requester's phone number. This field is present only if a phone number is available. For a punchout initiated from Oracle Exchange, this is the Exchange buyer's phone number.

<userEmail> Optional

Oracle iProcurement requester's e-mail address. For a punchout initiated from Oracle Exchange, this is the Exchange buyer's e-mail address.

<appUserName> Required

Oracle Applications user login name for the Oracle iProcurement requester (for example, TGREEN). For a punchout initiated from Oracle Exchange, this is the Exchange buyer's user name.

<companyName> Optional

In a punchout from Oracle iProcurement, this is the company name that is trying to log in to the supplier site and that was entered in the Company Name field while defining the punchout. In a punchout from Oracle Exchange, this is the company name on the Exchange.

<companyDUNS> Optional

The buying company's identifier entered in the Company ID field when defining the punchout in Oracle iProcurement. In a punchout initiated from Oracle Exchange, this field contains the DUNS number entered during registration, if any.

<contactName> Optional

In a loginRequest to Oracle Exchange, this field is not used.

In a loginRequest from Oracle Exchange to a supplier, this is the Exchange Company Administrator's name.

This field is not used for Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML).

<contactPhone> Optional

In a loginRequest to Oracle Exchange, this field is not used.

In a loginRequest from Oracle Exchange to a supplier, this is the Exchange Company Administrator's phone number.

This field is not used for Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML).

<returnURL> Required

In a loginRequest to Oracle Exchange, this is the URL in Oracle iProcurement to which Oracle Exchange will ask the browser to post the shopping cart.

In a loginRequest to the supplier, this is the URL in Oracle iProcurement or Oracle Exchange to which the supplier site will ask the browser to post the shopping cart.

In a return to Oracle iProcurement, the returnURL is constructed as follows, where APPS_FRAME_WORK_AGENT is the value in the *Application Framework Agent* profile option:

```
APPS_FRAME_WORK_AGENT + OA_HTML_DIRECTORY + OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_
CALLBACK&OAHP=<home page menu>&OASF=ICX_CAT_PUNCHOUT_
CALLBACK&transactionid=<current OADBTransaction ID>
```

For example:

```
http://qapache.us.oracle.com:15671/OA_HTML/OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_
CALLBACK&OAHP=ICX_POR_HOMEPAGE_MENU&OASF=ICX_CAT_PUNCHOUT_
CALLBACK&transactionid=1577779317
```

In a return to Oracle Exchange, the return URL is constructed as follows, where the first part of the URL (https://exchange.oracle.com) is the URL of the specific Exchange:

```
https://exchange.oracle.com/orders/PunchoutCallback.jsp
```

<partySiteId> Optional

Exchange Trading Partner ID of the supplier being punched out to, used only for a punchout via Oracle Exchange.

<searchKeywords> Optional

If the Oracle iProcurement requester or Oracle Exchange buyer searched for a particular set of key words prior to the punchout, these keywords are included here. This field is present only if the requester performed a search before accessing the punchout.

In a loginRequest to Oracle Exchange, if the requester already conducted a search with keywords in Oracle iProcurement and clicked the punchout link on the **Search Results** page, Oracle Exchange re-executes the search with those keywords and presents its **Search Results** page.

<cxmlCart> Optional

cXML shopping cart contents, used in Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML) only. (When Oracle iProcurement sends the cart back to Oracle Exchange for conversion, it does so via the loginRequest.)

<icxSessionCallBackURL> Optional

URL of the servlet that accesses the Oracle iProcurement session to keep the session alive (not idle) while the requester is shopping on the external site.

If the option *Extend user's idle session timeout during punchout* was selected during the punchout setup, then Oracle iProcurement populates this field as follows, where APPS_FRAME_WORK_AGENT is the value in the *Application Framework Agent* profile option:

```
APPS_FRAME_WORK_AGENT + OA_HTML_DIRECTORY + OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_
CALLBACK&OAHP=<home page menu>&OASF=ICX_CAT_PUNCHOUT_
HEARTBEAT&transactionid=<current OADBTransaction ID>
```

For example:

```
http://qapache.us.oracle.com:15671/OA_HTML/OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_
CALLBACK&OAHP=ICX_POR_HOMEPAGE_MENU&OASF=ICX_CAT_PUNCHOUT_
HEARTBEAT&transactionid=1577779317
```

In Model 1: Punchout from Oracle iProcurement to Oracle Exchange (XML), Oracle Exchange makes use of this field to extend the idle session. In Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML), the supplier must make use of this field to extend the idle session. Other models do not use this field.

* Optional extended data sent to the supplier if this option was chosen while defining the punchout in Oracle iProcurement. This field is used by Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML).

Changing vs. Unchanging Fields

The supplier uses the loginRequest to authenticate the user. As part of the authentication, the supplier often needs to know which fields in the loginRequest vary and which do not. For example, the password never changes for the same punchout. Therefore, the supplier knows that it will always authenticate the same password value for anyone who uses that punchout.

The following table summarizes the fields that can vary by user and the fields that never vary in a single punchout.

Table B–9   Changing vs. Unchanging Fields in loginRequest

Field Name	Varies by User	Typically Used for Authentication, if Desired
username	No	Yes
password	No	Yes
action	No	No
language	Yes	No
searchLanguage	Yes	No
operatingUnit	Yes, but stays the same within an operating unit	No
shipTo	Yes	No
deliverTo	Yes	No
fullName	Yes	No
title	Yes	No
manager	Yes	No
managerEmail	Yes	No
location	Yes	No
language	Yes	No
currency	Yes	No
dateFormat	Yes	No
exchangeName	No	No
userName	Yes	Yes
userPhone	Yes	No
userEmail	Yes	No
appUserName	Yes	Yes
companyName	No	Yes
companyDUNS	No	Yes
contactName	No	No
contactPhone	No	No
returnURL	No	No

Table B–9   Changing vs. Unchanging Fields in loginRequest

Field Name	Varies by User	Typically Used for Authentication, if Desired
partySiteId	No	No
searchKeywords	Yes	No
cxmlCart	Yes	No
icxSessionCallBackURL	No	No

loginResponse

Oracle Exchange or the supplier sends the loginResponse document in response to the punchout request in all models.

Example loginResponse

Below are example loginResponse documents.

Successful Connection   The following example loginResponse document shows a successful connection:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<response>
   <header version="1.0">
      <return returnCode="S"/>
   </header>
   <body>
      <loginURL>
<![CDATA[https://exchange.oracle.com/orders/LinkinCallback.jsp?sessionKey=84vw2w
nuq1.ml0Tah9NrkSIrlaIpR9vmQLz/AbJphDGpQbvp6vJqReUbxaPaK--1733&action=shopping&la
nguage=US&searchKeywords=]]>
      </loginURL>
   </body>
</response>
```

Unsuccessful Connection   The following example loginResponse document shows that an error, such as an invalid password, occurred during the connection:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<response>
   <header version="1.0">
      <return returnCode="A">
         <returnMessage>
```

```
          <![CDATA[Authentication Failure]]>
        </returnMessage>
      </return>
    </header>
    <body>
      <loginURL/>
    </body>
</response>
```

Field Descriptions

The fields in the loginResponse document are described below.

<returnMessage> Optional

Failure message, such as "Password Incorrect."

<return returnCode=""> Required

Valid values include S for *success*, A for *authentication* error (such as invalid password), E for *error*, U for *unexpected*, and W for *warning*.

<loginURL> Required if the connection was successful

URL to which the buyer is directed for shopping. As part of the validation process, a session should have been created that will be used to identify the buyer, and this session should be incorporated into this URL as shown in the successful example above.

shoppingCart

The shoppingCart XML document returns the shopping cart information to Oracle iProcurement or Oracle Exchange in all models.

Example shoppingCart

The following are example shoppingCart documents.

shoppingCart from Oracle Exchange to Oracle iProcurement The following is an example shoppingCart document that is sent to Oracle iProcurement from Oracle Exchange:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<response>
   <header version="1.0">
     <return returnCode="S"/>
```

```
        </header>
        <body>
            <OrderLinesDataElements>
                <catalogTradingPartner>
                    <![CDATA[OracleExchange]]>
                </catalogTradingPartner>
                <orderLine>
                    <contract>
                        <buyerContract>
                            <contractNumber>
                                <![CDATA[2345]]>
                            </contractNumber>
                        </buyerContract>
                        <catalogType>
                            <![CDATA[CONTRACTED]]>
                        </catalogType>
                    </contract>
                    <item lineType="GOODS" quantity="3.0">
                        <itemNumber>
                            <supplierItemNumber>
                                <itemID>
                                    <![CDATA[P456]]>
                                </itemID>
                            </supplierItemNumber>
                        </itemNumber>
                        <itemDescription>
                            <![CDATA[Red Pencil]]>
                        </itemDescription>
                        <unitOfMeasure>
                            <supplierUnitOfMeasure>
                                <supplierUOMType>
                                    <![CDATA[DZ]]>
                                </supplierUOMType>
                                <supplierUOMQuantity>
                                    <![CDATA[]]>
                                </supplierUOMQuantity>
                            </supplierUnitOfMeasure>
                        </unitOfMeasure>
                    </item>
                    <category>
                        <categoryCode categoryCodeIdentifier="SUPPLIER">
                            <![CDATA[BC Personal Digital Assistants (PDA's) or Pen-Based
Computers]]>
                        </categoryCode>
                    </category>
```

```
                    <price>
                       <currency>
                          <![CDATA[USD]]>
                       </currency>
                       <unitPrice>
                          <![CDATA[9.99]]>
                       </unitPrice>
                    </price>
                    <supplier>
                       <supplierTradingPartnerCode>
                          <![CDATA[105676]]>
                       </supplierTradingPartnerCode>
                       <supplierName>
                          <![CDATA[Supplier Corp.]]>
                       </supplierName>
                    </supplier>
                 </orderLine>
             </OrderLinesDataElements>
          </body>
</response>
```

shoppingCart from Supplier to Oracle iProcurement, with Configuration Number The
following is an example shoppingCart document that is sent to Oracle iProcurement
from the supplier, with a configuration number in the <supplierReferenceNumber>
field:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<response>
   <header version="1.0">
     <return returnCode="S"/>
   </header>
   <body>
      <OrderLinesDataElements>
         <catalogTradingPartner>Supplier A</catalogTradingPartner>
         <orderLine>
           <item lineType="GOODS" quantity="3.0">
               <itemNumber>
                  <supplierItemNumber>
                     <itemID>P456</itemID>
                    <supplierReferenceNumber>123123013021
               </supplierReferenceNumber>
                  </supplierItemNumber>
               </itemNumber>
               <itemDescription>Red Pencil</itemDescription>
               <unitOfMeasure>
```

```
                    <supplierUnitOfMeasure>
                        <supplierUOMType>DZ</supplierUOMType>
                        <supplierUOMQuantity/>
                    </supplierUnitOfMeasure>
                </unitOfMeasure>
            </item>
            <category>
                <categoryCode categoryCodeIdentifier="SUPPLIER">
BC Personal Digital Assistants (PDA's) or Pen-Based Computers</categoryCode>
            </category>
            <price>
                <currency>USD</currency>
                <unitPrice>9.99</unitPrice>
            </price>
            <supplier>
                <supplierTradingPartnerCode>105676
</supplierTradingPartnerCode>
                <supplierName>Supplier Corp.</supplierName>
            </supplier>
        </orderLine>
    </OrderLinesDataElements>
    </body>
</response>
```

Field Descriptions

The fields in the shoppingCart document are described below.

<return returnCode=""> Required

Valid values include S for *success*, A for *authentication* error (such as invalid password), E for *error*, U for *unexpected*, and W for *warning*.

<catalogTradingPartner> Required in a punchout from Oracle iProcurement, if <supplierName> is not populated

The <catalogTradingPartner> field is used by the following models, only if the buyer did not edit the Key 1 field after downloading the punchout definitions from Oracle Exchange; by default, the Key 1 field is left blank in these models:

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

If the buyer did not edit the Key 1 field after the download, then the value in the <catalogTradingPartner> field is matched to the Key 1 field in Oracle e-Commerce Gateway to perform category and UOM mapping. If the buyer did edit the Key 1 field after the download, then the value in that Key 1 field is matched to the Key 1 field in Oracle e-Commerce Gateway to perform the mapping, and the value in <catalogTradingPartner> is ignored.

This field is also used in Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML), for buyers who upgrade to this release. For upgrade customers, Oracle iProcurement uses the value in <catalogTradingPartner> to update the new, required Key 1 field on the punchout setup page.

<contract> Optional

Beginning of the contract section, which contains the <contractNumber>, <buyerContractLineNumber>, and <catalogType> fields. The <contract> field can contain an optional attribute, contractNumberIdentifier, with any of the following values: KNOWN, UNKNOWN, INFORMATIONAL, or NONE.

<contractNumber> Optional

Supplier or buyer contract number, depending on the parent field. This field is contained in the <contract> section. If this contract number matches an approved, effective contract purchase agreement number in Oracle Purchasing, the purchase order that ultimately gets created for the item will get created against that contract. To map the contract number to a contract purchase agreement in Oracle Purchasing, the parent field must be <buyerContract>. For a full description of the flow, buyers should see the Contract Numbers overview provided in the buyer version of this guide.

This field is not used in Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML).

<buyerContractLineNumber> Optional

Not used.

<catalogType> Optional

Indicator of whether the item comes from a contract catalog or non-contract catalog. If a buyer contractNumber is included, this field should be CONTRACTED or omitted. (If this field is not CONTRACTED or omitted, the contractNumber is ignored.) Oracle Exchange always sends a value of CONTRACTED in this field.

<item lineType="GOODS" quantity = "1.0"> Required

Beginning of the item section. The <item> field contains the following attributes:

- lineType. Possible values are GOODS, SERVICES_AMOUNT, or SERVICES_ QUANTITY. GOODS is the default when nothing is specified.

- quantity. Number of items ordered. The default is 1. Decimals are allowed.

<itemID> Required

Supplier, buyer, or manufacturer item number, depending on the parent field.

<supplierReferenceNumber> Optional

If the supplier specifies a <supplierReferenceNumber>, then that value is used as the configuration number for the item.

If the supplier provides a configuration number in this field, then this number travels with the requisition in Oracle iProcurement and with the purchase order in Oracle Purchasing. For more information, see Configurations and Re-Punchout on page 1-16.

<manufacturerName> Optional

Name of the manufacturer for this item. This name populates the Manufacturer Name in Oracle iProcurement.

<buyerItemRevision> Optional

Item revision number, contained in the <buyerItemNumber> field.

<itemDescription> Required

Supplier's description of this product. This description populates the Item Description in the Oracle iProcurement shopping cart and requisition.

<buyerUnitOfMeasure> Optional

Buyer's unit of measure code.

<supplierUOMType> Required

Supplier's unit of measure code. This code is used by the UOM mapping set up in Oracle e-Commerce Gateway.

<supplierUOMQuantity> Optional

Supplier's quantity in the specified unit of measure. For example, a Box of 12 would indicate Box for the <supplierUOMType> and 12 for the <supplierUOMQuantity>. The <supplierUOMQuantity>, if provided, is used by the UOM mapping that is set up in Oracle e-Commerce Gateway.

<hazardClass> Optional

Hazard class name for the item, if provided by the supplier.

<category categoryCodeIdentifier="SPSC"> Required

Category used to classify the item. Valid attribute code values are SPSC, SUPPLIER, or BUYER. If a code is not specified, SPSC is assumed.

<categoryCode> Required

Category code value. This code is used by the ITEM_CATEGORY (category) mapping that is set up in Oracle e-Commerce Gateway.

<currency> Required

Currency in which the price is specified.

<unitPrice> Required

The price per unit of the item.

<supplierDUNS> Conditionally Required

Supplier's DUNS number, used to retrieve the supplier name and supplier site information in Oracle iProcurement, only if the buyer did not specify a Supplier and Supplier Site during the punchout setup. Either a <supplierDUNS> or <supplierTradingPartnerCode> must be provided.

<supplierTradingPartnerCode> Conditionally Required

Either a <supplierDUNS> or <supplierTradingPartnerCode> must be provided. The <supplierTradingPartnerCode> is used if the supplier does not have a DUNS number. This field should be populated with a unique alphanumeric code that the buyer uses to identify the internal supplier code in Oracle Applications, for mapping, only if the buyer did not specify a Supplier and Supplier Site during the punchout setup.

<supplierName> Required

Supplier's company name. This field is used by Oracle iProcurement only if the buyer did not select a Supplier on the punchout setup page. If the buyer did select a Supplier, then this field is ignored.

The supplier name in this field is used by Oracle iProcurement for mapping to the VENDOR_NAME value in Oracle e-Commerce Gateway only if both of the following conditions are met:

- The buyer did not select a Supplier on the punchout setup page.

- The EDI Location field does not match the <supplierDUNS> or <supplierTradingPartnerCode>.

This field is also used in Model 2a: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (XML), for buyers who upgrade to this release. For upgrade customers, Oracle iProcurement uses the value in <supplierName> to update the new, required Key 1 field on the punchout setup page, if a <catalogTradingPartner> is not provided.

<supplierSite> Optional

Supplier site name. This field is used by Oracle iProcurement only if the buyer did not select a Supplier and Supplier Site on the punchout setup page. If the buyer did select a Supplier and Supplier Site, then this field is ignored.

The supplier site name in this field is used by Oracle iProcurement for mapping to the VENDOR_SITE External 2 value in Oracle e-Commerce Gateway only if both of the following conditions are met:

- The buyer did not select a Supplier and Supplier Site on the punchout setup page.

- The EDI Location field does not match the <supplierDUNS> or <supplierTradingPartnerCode>.

<contactName> Optional

Contact name for the supplier.

<contactPhone> Optional

Supplier contact's phone number.

<attribute1>...<attribute15> Optional

Up to 15 additional attributes can be included with each item in the shopping cart. These attributes travel with the requisition.

supplierSync

Oracle Exchange sends the supplierSynch document to Oracle iProcurement when the buying company downloads punchout suppliers from Oracle Exchange for the following models:

- Model 4: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (XML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

Example supplierSync

The following is an example supplierSync document:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<response>
   <header version="1.0">
     <return returnCode="S"/>
   </header>
   <body>
      <supplier>
         <supplierPartyId>
            <![CDATA[9176]]>
         </supplierPartyId>
         <supplierImageUrl>
            <![CDATA[http://www.mysite.com/logo.gif]]>
         </supplierImageUrl>
         <supplierLanguageSpecificInfo>
            <language>
               <![CDATA[US]]>
            </language>
            <supplierName>
               <![CDATA[Staples]]>
            </supplierName>
            <supplierDescription>
               <![CDATA[]]>
            </supplierDescription>
            <supplierKeywords>
               <![CDATA[]]>
```

```
            </supplierKeywords>
          </supplierLanguageSpecificInfo>
        </supplier>
      </body>
</response>
```

Field Descriptions

The fields in the SupplierSyncUpResponse document are described below.

<supplierPartyId> Required

Supplier's Trading Partner ID on Oracle Exchange.

<supplierImageUrl> Required

URL pointing to the supplier's logo (entered by the supplier on Oracle Exchange).

<language> Required

Language of the information being retrieved, using the Oracle Applications language code.

<supplierName> Required

Supplier's company name in the specified language (the Exchange registered name of the company).

<supplierDescription> Required

Supplier's punchout definition description in the specified language (entered by the supplier on Oracle Exchange).

<supplierKeywords> Required

Keywords provided by the supplier when defining the punchout on Oracle Exchange, in the specified <language>.

PunchOutSetupRequest

Oracle Exchange sends the PunchOutSetupRequest document to the supplier in the following models:

- Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML)

- Model 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (cXML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

PunchOutSetupRequest.dtd
See the cXML DTD at http://www.cxml.org/.

Example PunchOutSetupRequest
The following are example PunchOutSetupRequest documents.

PunchOutSetupRequest from Oracle iProcurement to Supplier Directly The following is an example PunchOutSetupRequest document sent from Oracle iProcurement to the supplier directly:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US"
payloadID="20040316032452.913060910.144270@ap6172rt.us.oracle.com"
timestamp="2004-03-16T03:24:52-08:00">
   <Header>
      <From>
         <Credential domain="DUNS">
            <Identity>123456789</Identity>
         </Credential>
      </From>
      <To>
         <Credential domain="DUNS">
            <Identity>987654321</Identity>
         </Credential>
      </To>
      <Sender>
         <Credential domain="DUNS">
            <Identity>123456789</Identity>
            <SharedSecret>welcome</SharedSecret>
         </Credential>
         <UserAgent>Oracle iProcurement</UserAgent>
      </Sender>
   </Header>
   <Request>
      <PunchOutSetupRequest operation="create">
         <BuyerCookie>12345678</BuyerCookie>
         <Extrinsic name="User">OPERATIONS</Extrinsic>
```

```
        <BrowserFormPost>
            <URL>http://qapache.us.oracle.com:15671/OA_HTML/OA.jsp?OAFunc=ICX_
CAT_PUNCHOUT_CALLBACK&#38;OAHP=ICX_POR_HOMEPAGE_MENU&#38;OASF=ICX_CAT_PUNCHOUT_
CALLBACK&#38;transactionid=1577779317</URL>
        </BrowserFormPost>
        <Contact>
            <Name xml:lang="en-US">Stock, Pat</Name>
            <Email>pat.stock@vision.com</Email>
        </Contact>
        <SupplierSetup>
            <URL>http://abc.com/Gateway/Company/Login.jsp</URL>
        </SupplierSetup>
    </PunchOutSetupRequest>
  </Request>
</cXML>
```

PunchOutSetupRequest from Oracle Exchange to Supplier The following is an example
PunchOutSetupRequest document sent from Oracle Exchange to the supplier
directly:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US"
payloadID="Tue Mar 16 23:24:19 PST 2004" timestamp="Tue Mar 16 23:24:19 PST
2004">
  <Header>
    <From>
        <Credential domain="DUNS">
            <Identity>123456789</Identity>
        </Credential>
    </From>
    <To>
        <Credential domain="DUNS">
            <Identity>987654321</Identity>
        </Credential>
    </To>
    <Sender>
        <Credential domain="Oracle Exchange">
            <Identity>exchange.oracle.com</Identity>
            <SharedSecret>welcome</SharedSecret>
        </Credential>
        <UserAgent>Oracle Exchange</UserAgent>
    </Sender>
  </Header>
  <Request>
```

```
      <PunchOutSetupRequest operation="create">
         <BuyerCookie>12345678</BuyerCookie>
         <Extrinsic name="User">LIWANG</Extrinsic>
         <BrowserFormPost>
<URL>https://testexchange.oracle.com/orders/PunchoutCallback.jsp</URL>
         </BrowserFormPost>
         <Contact>
            <Name xml:lang="en-US">LiWang</Name>
            <Email>liwang@oracle.com</Email>
         </Contact>
         <SupplierSetup>
            <URL>http://abc.com/Gateway/Company/Login.jsp</URL>
         </SupplierSetup>
      </PunchOutSetupRequest>
   </Request>
</cXML>
```

PunchOutSetupRequest in a Re-Punchout The following is an example
PunchOutSetupRequest document sent during a re-punchout, in Model 2b:
Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML):

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE cXML SYSTEM "http://xml.cxml.org/schemas/cXML/1.1.007/cXML.dtd">
<cXML version="1.1.007" xml:lang="en-US"
payloadID="20040316032452.913060910.144270@ap6172rt.us.oracle.com"
timestamp="2004-03-16T03:24:52-08:00">
   <Header>
      <From>
         <Credential domain="DUNS">
            <Identity>123456789</Identity>
         </Credential>
      </From>
      <To>
         <Credential domain="DUNS">
            <Identity>987654321</Identity>
         </Credential>
      </To>
      <Sender>
         <Credential domain="Oracle Exchange">
            <Identity>exchange.oracle.com</Identity>
            <SharedSecret>welcome</SharedSecret>
         </Credential>
         <UserAgent>Oracle Exchange</UserAgent>
      </Sender>
   </Header>
```

```
        <Request>
           <PunchOutSetupRequest operation="inspect">
               <ItemOut quantity="2">
                  <SupplierPartID>667</SupplierPartID>
                  <SupplierPartAuxID>12345</SupplierPartAuxID>
              </ItemOut>
              <BuyerCookie>12345678</BuyerCookie>
              <Extrinsic name="User">LIWANG</Extrinsic>
              <BrowserFormPost>
<URL>http://qapache.us.oracle.com:15671/OA_HTML/OA.jsp?OAFunc=ICX_CAT_PUNCHOUT_
CALLBACK&#38;OAHP=ICX_POR_HOMEPAGE_MENU&#38;OASF=ICX_CAT_PUNCHOUT_
CALLBACK&#38;transactionid=1577779317</URL>
              </BrowserFormPost>
              <Contact>
                 <Name xml:lang="en-US">LiWang</Name>
                 <Email>liwang@oracle.com</Email>
              </Contact>
              <SupplierSetup>
                 <URL>http://abc.com/Gateway/Company/Login.jsp</URL>
              </SupplierSetup>
           </PunchOutSetupRequest>
        </Request>
</cXML>
```

Field Descriptions

The following describes some of the fields in the PunchOutSetupRequest document. For complete information, see the *cXML User's Guide* available at http://www.cxml.org/.

<From> <Credential> Required

Identifier for the buying organization. The domain attribute is the value that the buyer entered in the Domain field while defining the punchout. The identity element is the value that the buyer entered in the Identity field while defining the punchout.

In a punchout from Oracle Exchange, the domain attribute will be either of the following:

- DUNS. If the company has a DUNS number on Oracle Exchange, the identity element will have the DUNS number.

- NAME. If there is no DUNS number, the identity element will have the name of the company on Oracle Exchange.

For Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML), the FromCredential is always the Exchange name and URL as shown in the following example:

```
<From>
 <Credential domain="exchange.oracle.com">
   <Identity>Oracle Exchange</Identity>
 </Credential>
 </From>
```

The Exchange URL is taken from the Exchange Operator **Software Setup** page, from the System URL field. The Exchange name is taken from the Exchange Operator **Identification** page, from the System Name field.

<To> <Credential> Required

Identifier for the supplier organization. The domain attribute will be either of the following:

- DUNS. In a cXML punchout from Oracle Exchange, the identity field will have the DUNS number if the company has a DUNS number on Oracle Exchange. In a cXML punchout to the supplier, DUNS is the Supplier ID on the punchout setup page, if entered.

- NAME. If there is no DUNS number, the identity field will have the name of the company on Oracle Exchange. In a cXML punchout to the supplier, the NAME is the Supplier Name on the punchout setup page.

<Sender> <Credential> Required

In a punchout from Oracle Exchange, the domain attribute identifies the Exchange name. The identity field gives the Exchange site URL, such as exchange.oracle.com. In a punchout from Oracle iProcurement, the <Sender><Credential> values are the same as the <From><Credential> values.

The shared secret field gives the password validated by the supplier site.

<Sender> <UserAgent> Required

Name of the Exchange, such as Oracle Exchange.

<PunchOutSetupRequest operation="create"> Required

Oracle iProcurement supports only "inspect" or "create" options in this attribute, not the "edit" option. In the initial punchout, this attribute is set to "create."

If the supplier specified "inspect" in the PunchOutOrderMessage document, then a re-punchout from Oracle iProcurement back to the supplier site is allowed. In the re-punchout, Oracle iProcurement sends the PunchOutSetupRequest with this attribute set to "inspect." If the supplier specified "create" in the PunchOutOrderMessage document, then a re-punchout is not allowed. The supplier could specify "edit"; however, Oracle iProcurement does not support editing configuration details during a re-punchout, only viewing them.

Re-punchout is used only by Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML).

For more information, see Re-punchout on page 1-21.

<Extrinsic name="User"> Required

User login name of the buyer on Oracle Exchange for Model 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (cXML); user login name of the Oracle iProcurement requester for Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML) and Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML).

<BrowserFormPost> <URL> Required

URL where the browser posts the shopping cart cXML document after the supplier sends the shopping cart to the browser.

<Contact> Optional

The buyer name and e-mail fields within the contact element are passed to the supplier.

Changing vs. Unchanging Fields

The supplier uses the PunchOutSetupRequest to authenticate the user. As part of the authentication, the supplier often needs to know which fields in the request vary and which do not. For example, the From Credential never changes for the same punchout. Therefore, the supplier knows that it will always authenticate the same From Credential value for anyone who uses that punchout.

The following table summarizes some of the fields that can vary by user and the fields that never vary in a single punchout.

Table B–10   Changing vs. Unchanging Fields in PunchOutSetupRequest

Field Name	Varies by User	Typically Used for Authentication, if Desired
From Credential	No	Yes
To Credential	No	No
Sender Credential	No	Yes
Sender SharedSecret	No	Yes
Sender UserAgent	No	No
Extrinsic name="User"	Yes	Yes
URL	No	No
Contact	Yes	No

PunchOutSetupResponse

The cXML supplier sends the PunchOutSetupResponse document to Oracle Exchange in the following models:

- Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML)

- Model 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (cXML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

PunchOutSetupResponse.dtd

See the cXML DTD at http://www.cxml.org/.

Example PunchOutSetupResponse

Below are example PunchOutSetupResponse documents.

Successful Connection The following example document shows a successful connection:

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE cXML SYSTEM
"http://xml.cxml.org/schemas/cXML/1.1.010/cXML.dtd"><cXML version="1.1.007"
xml:lang="en-US" payloadID="200303450803006749@b2b.euro.com"
timestamp="2003-01-12T08:03:00">
<Response>
```

```
        <Status code="200" text="OK"/>
        <PunchOutSetupResponse>
            <StartPage>
<URL>https://abc.com/Gateway/Company/Login.jsp?SessionID=69F49ED9-9BE9-43AE-AD6B
-4F9E54FF7996</URL>
            </StartPage>
            </PunchOutSetupResponse>
        </Response>
</cXML>
```

Unsuccessful Connection The following example document shows that an error, such as an invalid password, occurred during the connection; the status code is a cXML status code:

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE cXML SYSTEM
"http://xml.cxml.org/schemas/cXML/1.1.010/cXML.dtd"><cXML version="1.1.007"
xml:lang="en-US" payloadID="200303450813247015@b2b.euro.com"
timestamp="2003-01-12T08:13:24">
    <Response>
        <Status code="401" text=""/>
    </Response>
</cXML>
```

Field Descriptions

For descriptions of the fields in the PunchOutSetupResponse document, see the *cXML User's Guide* available at http://www.cxml.org/.

PunchOutOrderMessage

The PunchOutOrderMessage document returns the shopping cart information to Oracle Exchange in cXML format in the following models:

- Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML)

- Model 3b: Punchout from Oracle Exchange to Supplier-Hosted Catalog (cXML)

- Model 5: Punchout from Oracle iProcurement to Supplier-Hosted Catalog via Oracle Exchange (cXML)

PunchOutOrderMessage.dtd

See the cXML DTD at http://www.cxml.org/.

Example PunchOutOrderMessage

The following is an example PunchOutOrderMessage document:

```
<?xml version="1.0" encoding="UTF-8"?>
<cXML payloadID="20030345080932789@euro.com" timestamp="2003-01-12%2008:09:32">
<Header>
     <From>
          <Credential domain="DUNS">
               <Identity>987654321</Identity>
          </Credential>
     </From>
     <To>
          <Credential domain="DUNS">
               <Identity>123456789</Identity>
          </Credential>
     </To>
     <Sender>
          <Credential domain="DUNS">
               <Identity>987654321</Identity>
          </Credential>
<UserAgent/>
     </Sender>
</Header>
<Message>
<PunchOutOrderMessage>
     <BuyerCookie>12345678</BuyerCookie>
     <PunchOutOrderMessageHeader operationAllowed="create">
          <Total>
               <Money currency="GBP">924.00</Money>
          </Total>
     </PunchOutOrderMessageHeader>
     <!-- BASE ITEM  -->
     <ItemIn quantity="1">
          <ItemID>
               <SupplierPartID>2041711,39630</SupplierPartID>
               <SupplierPartAuxiliaryID>12345</SupplierPartAuxiliaryID>
          </ItemID>
          <ItemDetail>
               <UnitPrice>
                    <Money currency="GBP">899.00</Money>
               </UnitPrice>
               <Description xml:lang="en-US">Jan 1 Promotion - Mainstream
Mobility (C640) 200-17009</Description>
               <UnitOfMeasure>EA</UnitOfMeasure>
               <Classification domain="SPSC">43000000</Classification>
```

```
                            <Classification domain="UNSPSC">43000000</Classification>
                            <ManufacturerPartID>39630</ManufacturerPartID>
                            <ManufacturerName>Manufacturer Corporation</ManufacturerName>
                    </ItemDetail>
            </ItemIn>
            <!-- DELIVERY -->
            <ItemIn quantity="1">
                    <ItemID>
                            <SupplierPartID>2041711</SupplierPartID>
                            <SupplierPartAuxiliaryID/>
                    </ItemID>
                    <ItemDetail>
                            <UnitPrice>
                                    <Money currency="GBP">25.00</Money>
                            </UnitPrice>
                            <Description xml:lang="en-US">Delivery</Description>
                            <UnitOfMeasure>EA</UnitOfMeasure>
                            <Classification domain="SPSC">43000000</Classification>
                            <Classification domain="UNSPSC">43000000</Classification>
                            <ManufacturerPartID/>
                            <ManufacturerName/>
                    </ItemDetail>
            </ItemIn>
</PunchOutOrderMessage>
</Message>
</cXML>
```

Field Descriptions

The following describes some of the fields in the PunchOutOrderMessage document,. For complete information, see the *cXML User's Guide* available at http://www.cxml.org/.

<PunchOutOrderMessageHeader operationAllowed="create"> Required

Oracle iProcurement supports only "inspect" or "create" options in this field attribute, not the "edit" option. If the supplier specifies "inspect," then a re-punchout from Oracle iProcurement back to the supplier site is allowed. If the supplier specifies "create," then a re-punchout is not allowed. The supplier can specify "edit"; however, Oracle iProcurement treats "edit" like the "inspect" option. Oracle iProcurement does not support editing configuration details during a re-punchout, only viewing them.

Re-punchout is used only by Model 2b: Punchout from Oracle iProcurement Directly to Supplier-Hosted Catalog (cXML).

For more information, see Re-punchout on page 1-21.

<SupplierPartAuxiliaryID> Optional

If the supplier specifies a <SupplierPartAuxiliaryID>, then that value is used as the configuration number for the item.

If the supplier provides a configuration number in this field, then this number travels with the requisition in Oracle iProcurement and with the purchase order in Oracle Purchasing. For more information, see Configurations and Re-Punchout on page 1-16.

ItemSearchRequest

Oracle iProcurement uses the ItemSearchRequest document to conduct a transparent punchout on Oracle Exchange or a supplier site, depending on the transparent punchout model used:

- Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)

- Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)

ItemSearchRequest.dtd

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Version 1.0 of Oracle Transparent Punchout Item Search Request DTD
     User by Oracle Exchange 6.2.4 and Oracle iProcurement Release FPI
 -->

<!-- Start Item Search Request DTD -->

<!ELEMENT DistributedSearchXML(Header,Request)>
  <!ATTLIST DistributedSearchXML payloadID CDATA #REQUIRED>
  <!ELEMENT Header(From,To,Login)>
    <!ATTLIST Header version CDATA #REQUIRED>
    <!ELEMENT From (Credential)>
    <!ELEMENT To (Credential)>
      <!ELEMENT Credential (Identity)>
      <!ATTLIST Credential domain CDATA #REQUIRED>
```

```
            <!ELEMENT Identity(#PCDATA)>
       <!ELEMENT Login (UserName,Password,AuthenticatedKey?)>
         <!ELEMENT UserName (#PCDATA)>
         <!ELEMENT Password (#PCDATA)>
         <!ELEMENT AuthenticatedKey (#PCDATA)>
     <!ELEMENT Request (ItemSearchRequest)>
       <!ELEMENT ItemSearchRequest (UserInfo?,SearchInfo,UserArea?)>
       <!ATTLIST ItemSearchRequest operation CDATA #IMPLIED>
         <!ELEMENT UserInfo (UserName?,AppsUserName?,UserPhone?,UserEmail?)>
           <!ELEMENT AppsUserName (#PCDATA)
           <!ELEMENT UserPhone (#PCDATA)>
           <!ELEMENT UserEmail (#PCDATA)>
         <!ELEMENT SearchInfo
(SearchLanguage,SearchKeywords,SortBy?,ResultSize,StartResult)>
           <!ELEMENT SearchLanguage (#PCDATA)>
           <!ELEMENT SearchKeywords (#PCDATA)>
           <!ELEMENT SortBy (#PCDATA)>
           <!ATTLIST SortBy order (ASC|DESC) #REQUIRED>
           <!ELEMENT ResultSize (#PCDATA)>
           <!ELEMENT StartResult (#PCDATA)>
         <!ELEMENT UserArea (SupplementalInfo*)>
           <!ELEMENT SupplementalInfo (#PCDATA)>
           <!ATTLIST SupplementalInfo name CDATA #REQUIRED>

<! --- End Item Search Request DTD -->
```

Example ItemSearchRequest

The following is an example ItemSearchRequest document:

```
<?xml version="1.0" encoding="utf-8"?>
<DistributedSearchXML payloadID = "123465678">
   <Header version="1.0">
      <From>
         <Credential domain="DUNS">
            <Identity>123456789</Identity>
         </Credential>
      </From>
      <To>
         <Credential domain="Name">
            <Identity>Disneyland</Identity>
         </Credential>
      </To>
      <Login>
         <UserName>operations</UserName>
         <Password>welcome</Password>
```

```
            <AuthenticatedKey>12345678</AuthenticatedKey>
        </Login>
    </Header>
    <Request>
        <ItemSearchRequest operation="SimpleSearch">
            <UserInfo>
                <UserName></UserName>
                <AppsUserName>ppan</AppsUserName>
                <UserPhone></UserPhone>
                <UserEmail></UserEmail>
            </UserInfo>
            <SearchInfo>
                <SearchLanguage>EN-US</SearchLanguage>
                <SearchKeywords>blue pen</SearchKeywords>
                <SortBy order="DESC">Price</SortBy>
                <ResultSize>15</ResultSize>
                <StartResult>1</StartResult>
            </SearchInfo>
            <UserArea>
                <SupplementalInfo name="Group">OU1</SupplementalInfo>
                <SupplementalInfo name="Division">NorthEast</SupplementalInfo>
            </UserArea>
        </ItemSearchRequest>
    </Request>
</DistributedSearchXML>
```

Field Descriptions

The fields in the ItemSearchRequest document are described below.

<From> <Credential> Required

Identifier for the buying organization. The domain attribute will be what was entered in the Domain field while defining the transparent punchout. The identity element is what was entered in the Identity field while defining the transparent punchout.

<To> <Credential> Required

Identifier for the supplier organization. If conducting a transparent punchout to a supplier:

■   If only a Supplier Name (not a Supplier ID) was entered on the transparent punchout setup page, the domain attribute is NAME and the identity is the Supplier Name.

- If a Supplier ID was also entered on the transparent punchout setup page, the domain attribute is DUNS and the identity is the Supplier ID.

If conducting a transparent punchout to Oracle Exchange, the ToCredential is always Oracle Exchange as shown in the following example:

```
<To>
 <Credential domain="">
   <Identity>Oracle Exchange</Identity>
 </Credential>
</To>
```

<Login> <UserName> Required

User name of the Oracle iProcurement requester used for authentication by the external site. For a transparent punchout to Oracle Exchange, this is the proxy user name assigned to the requester. For a transparent punchout to a supplier, the supplier needs to provide the user name to the buying organization. The user name comes from the buying organization's transparent punchout setup.

<Login> <Password> Required

Password of the Oracle iProcurement requester used for authentication by the external site. For a transparent punchout to Oracle Exchange, this is the proxy user password assigned to the requester. For a transparent punchout to a supplier, the supplier needs to provide the password to the buying organization. The password comes from the buying organization's transparent punchout setup.

> **Note:** The password is not encrypted in the XML document. It is up to the transport protocol to encrypt the message to ensure security.

<Login> <AuthenticatedKey> Optional

Key that Oracle iProcurement received from the external site in previous responses. It is recommended that the supplier provide the key for faster performance, since it avoids repeat logins.

<ItemSearchRequest> Required

The operation attribute in this field defines the type of search to perform. The currently supported value is SimpleSearch.

<UserInfo> <UserName> Optional extended data *

Full name of the Oracle iProcurement requester. This field is reserved for future use.

<UserInfo> <AppsUserName> Optional extended data *

Oracle Applications user login name for the Oracle iProcurement requester (for example, TGREEN).

<UserInfo> <userPhone> Optional extended data *

Oracle iProcurement requester's phone number. This field is reserved for future use.

<UserInfo> <userEmail> Optional extended data *

Oracle iProcurement requester's e-mail address. This field is reserved for future use.

<searchLanguage> Required

Requester's search language in Oracle iProcurement. The search language can be chosen by clicking the "Change Catalog Language" link on the **Shop** home page. (For example, if the session language is English, but catalog items exist only in German, the requester can choose German as the search language.)

The language uses the ISO 639 language and ISO 3166 territory code formats—for example, EN-US for United States English. The language code is EN, and the country code is US.

The supplier should honor the search language and return results in that language. Oracle Exchange returns results in the specified search language, if the supplier loaded item translations in that language to the Exchange catalog. (If no translations exist in that Exchange language, the search response informs the requester that no results were found.)

<searchKeywords> Required

Search string the requester entered. The search string may include an asterisk (*) if the requester performed a wildcard search. (If the requester used % instead of * to perform the wildcard search, Oracle iProcurement converts % to *.) Therefore, the supplier needs to interpret and support the * wildcard character. For example, the search string *ball** should return all items that start with *ball*. The supplier can interpret the rest of the search string however it wants. If there are multiple words in the search string, it is suggested that the supplier interpret the query matching condition as AND, to make the behavior consistent with Oracle iProcurement. For

example, if the search string is *legal paper*, Oracle iProcurement returns all items that contain the words *legal* AND *paper* in their searchable item information.

When conducting a transparent punchout to Oracle Exchange, Oracle Exchange supports the following search operators: *, %, -, and "". If these are included in the search string, Oracle Exchange will use them to conduct the search and return the relevant results.

<SortBy> Optional

Sort-by method. The order attribute indicates whether the sort is ascending (ASC) or descending (DESC). The value indicates what to sort by. The currently supported sort-by value is Price. Step "5 Oracle iProcurement displays search results" on page A-20 provides more information on this process.

<ResultSize> Required

Number of search results requested. This number is based on the requester's user profile (accessed in the requester's preferences), in the Shopping Search Results Per Page field. This number is two times the Shopping Search Results Per Page. For example, if the requester's Shopping Search Results Per Page preference is 7, the <ResultSize> is 14. Step "5 Oracle iProcurement displays search results" on page A-20 provides more information on this process.

<StartResult> Required

Index number to be assigned to the first search result. This number will be used for indexing when fetching additional search results. In the following example, the requester's Shopping Search Results Per Page preference is 7. The first time the request is sent, the <StartResult> is 1 and the <ResultSize> is 15 (two times 7, plus 1). Thereafter, the <ResultSize> remains 14 (two times 7), and the <StartResult> starts at the beginning of the next result set.

Request	<StartResult>	<ResultSize>
First request	1	15
Second request	16	14
Third request	30	14

Step "5 Oracle iProcurement displays search results" on page A-20 provides more information on this process.

<SupplementalInfo> Optional

Additional elements the buying organization can pass to the external site. The supplier can use this information to further identify the buyer and return search results specific to that buyer—for example, buyer-specific prices for a particular geographical region. Use the name attribute to identify the element, such as Division. Then provide the value in the field, such as NorthEast.

This field is not processed by Oracle Exchange.

* Optional user information sent to the supplier if this option was chosen while defining the transparent punchout in Oracle iProcurement. This field is used by Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML). (If optional user information was not chosen, the UserInfo section is not included.)

Changing vs. Unchanging Fields

The supplier uses the ItemSearchRequest to authenticate the user. As part of the authentication, the supplier often needs to know which fields in the request vary and which do not. For example, the From Credential never changes for the same transparent punchout. Therefore, the supplier knows that it will always authenticate the same From Credential value for anyone who uses that transparent punchout.

The following table summarizes some of the fields that can vary by user and the fields that never vary in a single transparent punchout.

Table B–11   Changing vs. Unchanging Fields in ItemSearchRequest

Field Name	Varies by User	Typically Used for Authentication, if Desired
From Credential	No	Yes
To Credential	No	No
Login UserName	No	Yes
Login Password	No	Yes
Login AuthenticatedKey	No	Yes
ItemSearchRequest attribute	No	No
UserInfo UserName	Yes	No

Table B–11   Changing vs. Unchanging Fields in ItemSearchRequest

Field Name	Varies by User	Typically Used for Authentication, if Desired
UserInfo AppsUserName	Yes	Yes
UserInfo userPhone	(Not used)	(Not used)
UserInfo userEmail	(Not used)	(Not used)
searchLanguage	Yes	No
searchKeywords	Yes	No
SortBy	Yes	No
ResultSize	Yes	No
StartResult	Yes	No
SupplementalInfo	No	Yes

ItemSearchResponse

Oracle iProcurement uses the ItemSearchResponse document to conduct a transparent punchout on Oracle Exchange or a supplier site, depending on the transparent punchout model used:

- Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML)

- Model 2: Transparent Punchout from Oracle iProcurement to Supplier-Hosted Catalog (XML)

ItemSearchResponse.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $Header: ItemSearchResponse_v1_0.dtd 115.1 2003/03/05 22:09:02 sthadaka
noship $ -->
<!-- Version 1.0 of Oracle Transparent Punchout Item Search Response DTD
     User by Oracle Exchange 6.2.4 and Oracle iProcurement Release FPI
 -->

<!-- Start Item Search Response DTD -->

<!ELEMENT DistributedSearchXML (Header,Response)>
  <!ELEMENT Header (From,To,AuthenticatedKey?)>
    <!ATTLIST Header version CDATA #REQUIRED>
    <!ELEMENT From (Credential)>
```

```
<!ELEMENT To (Credential)>
<!ELEMENT AuthenticatedKey (#PCDATA)>
  <!ELEMENT Credential (Identity)>
  <!ATTLIST Credential domain CDATA #REQUIRED>
    <!ELEMENT Identity (#PCDATA)>
<!ELEMENT Response (ItemSearchResponse)>
  <!ELEMENT ItemSearchResponse (SearchResultsInfo,Status,ItemInfo*)>
    <!ELEMENT SearchResultsInfo
(ResultsCount,TotalResultsCount?,SearchTime?,SortedBy?)>
      <!ELEMENT ResultsCount (#PCDATA)>
      <!ELEMENT TotalResultsCount (#PCDATA)>
      <!ELEMENT SearchTime (#PCDATA)>
      <!ELEMENT SortedBy (#PCDATA)>
      <!ATTLIST SortedBy order CDATA #REQUIRED>
    <!ELEMENT Status (#PCDATA)>
      <!ATTLIST Status code CDATA #REQUIRED>
    <!ELEMENT ItemInfo
(SupplierPartNum,SupplierInfo,Description,UnitPrice,UOMInfo,
ManufacturerPartNum?, ManufacturerName?, ThumbnailImage?, Image?,
CategoryInfo,CatalogType?,BuyerContract?,ItemAttribute*)>
      <!ATTLIST ItemInfo lineType #CDATA #REQUIRED>
      <!ELEMENT SupplierPartNum (#PCDATA)>
      <!ELEMENT SupplierInfo (SupplierCode,SupplierName)>
        <!ELEMENT SupplierCode (#PCDATA)>
        <!ELEMENT SupplierName (#PCDATA)>
      <!ELEMENT Description (ShortName,LongName?)>
        <!ELEMENT ShortName (#PCDATA)>
        <!ELEMENT LongName (#PCDATA)>
      <!ELEMENT UnitPrice (Money)>
        <!ELEMENT Money (#PCDATA)>
        <!ATTLIST Money currency CDATA #REQUIRED>
      <!ELEMENT UOMInfo (UOMCode,UOMName?)>
        <!ELEMENT UOMCode (#PCDATA)>
        <!ELEMENT UOMName (#PCDATA)>
      <!ELEMENT ManufacturerPartNum (#PCDATA)>
      <!ELEMENT ManufacturerName (#PCDATA)>
      <!ELEMENT ThumbnailImage (URL)>
      <!ELEMENT Image (URL)>
        <!ELEMENT URL (#PCDATA)>
      <!ELEMENT CategoryInfo (CategoryCode,CategoryName?)>
        <!ELEMENT CategoryCode (#PCDATA)>
        <!ELEMENT CategoryName (#PCDATA)>
      <!ELEMENT CatalogType (#PCDATA)>
      <!ELEMENT BuyerContract (#PCDATA)>
      <!ELEMENT ItemAttribute (#PCDATA)>
```

```
            <!ATTLIST ItemAttribute
             name CDATA #REQUIRED
             type CDATA #REQUIRED>

<! --- End Item Search Response DTD -->

<! --- End Distributed Search Response DTD -->
```

Example ItemSearchResponse

Below are example ItemSearchResponse documents.

Successful Response The following example ItemSearchResponse document shows a successful connection:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<DistributedSearchXML>
   <Header version="1.0">
      <From>
         <Credential domain="Name">
            <Identity>DisneyLand</Identity>
         </Credential>
      </From>
      <To>
         <Credential domain="DUNS">
            <Identity>123456789</Identity>
         </Credential>
      </To>
      <AuthenticatedKey>12345678</AuthenticatedKey>
   </Header>
   <Response>
      <ItemSearchResponse>
         <SearchResultsInfo>
            <ResultsCount>10</ResultsCount>
            <TotalResultsCount>234</TotalResultsCount>
            <SearchTime>0.75</SearchTime>
            <SortedBy order="desc">Price</SortedBy>
         </SearchResultsInfo>
         <Status code="200">Success</Status>
         <ItemInfo lineType="Goods">
            <SupplierPartNum>CR008521</SupplierPartNum>
            <SupplierInfo>
               <SupplierCode>78789797</SupplierCode>
               <SupplierName>CE Office Supply</SupplierName>
            </SupplierInfo>
```

```
            <Description>
               <ShortName>Blue pencil</ShortName>
               <LongName>Cross refills for Selectip Rollerball pens.</LongName>
            </Description>
            <UnitPrice>
               <Money currency="JPY">400</Money>
            </UnitPrice>
            <UOMInfo>
               <UOMCode>EA</UOMCode>
               <UOMName>EA</UOMName>
            </UOMInfo>
            <ManufacturerPartNum>CRO8521</ManufacturerPartNum>
            <ManufacturerName>BIC</ManufacturerName>
            <ThumbnailImage>
               <URL>http://www.image.com/thumbnailimage/5555.gif</URL>
            </ThumbnailImage>
            <Image>
               <URL>http://www.image.com/normalimage/5555.gif</URL>
            </Image>
            <CategoryInfo>
               <CategoryCode>44978878</CategoryCode>
               <CategoryName>Cartridge Pen Ink Refills</CategoryName>
            </CategoryInfo>
            <CatalogType>CONTRACTED</CatalogType>
            <BuyerContract>852</BuyerContract>
            <ItemAttribute name="Ink Color" type="Text">Blue</ItemAttribute>
            <ItemAttribute name="Model" type="Number">21</ItemAttribute>
            <ItemAttribute name="Point Style" type="Text">Fine</ItemAttribute>
         </ItemInfo>
      </ItemSearchResponse>
   </Response>
</DistributedSearchXML>
```

Unsuccessful Response The following example ItemSearchResponse document shows an unsuccessful connection:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<DistributedSearchXML>
   <Header version="1.0">
      <From>
         <Credential domain="Name">
            <Identity>DisneyLand</Identity>
         </Credential>
      </From>
      <To>
```

```
            <Credential domain="DUNS">
                <Identity>123456789</Identity>
            </Credential>
        </To>
        <AuthenticatedKey>12345678</AuthenticatedKey>
    </Header>
    <Response>
        <ItemSearchResponse>
            <SearchResultsInfo>
                <ResultsCount>0</ResultsCount>
            </SearchResultsInfo>
            <Status code="401">Authentication Failure</Status>
        </ItemSearchResponse>
    </Response>
</DistributedSearchXML>
```

No Results Found The following example ItemSearchResponse document occurs when no results are found that match the search criteria:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<DistributedSearchXML>
    <Header version="1.0">
        <From>
            <Credential domain="Name">
                <Identity>DisneyLand</Identity>
            </Credential>
        </From>
        <To>
            <Credential domain="DUNS">
                <Identity>123456789</Identity>
            </Credential>
        </To>
        <AuthenticatedKey>12345678</AuthenticatedKey>
    </Header>
    <Response>
        <ItemSearchResponse>
            <SearchResultsInfo>
                <ResultsCount>0</ResultsCount>
                <TotalResultsCount>0</TotalResultsCount>
                <SearchTime>0.75</SearchTime>
                <SortedBy order="desc"></SortedBy>
            </SearchResultsInfo>
            <Status code="200">Success</Status>
        </ItemSearchResponse>
    </Response>
```

```
</DistributedSearchXML>
```

Field Descriptions

The fields in the ItemSearchResponse document are described below.

<From> <Credential> Required

Identifier for the supplier organization. The recommendation is to use the same values the buying company sent in the <To><Credential> fields in the ItemSearchRequest document. Oracle Exchange use the same values the buying company sent in the <To><Credential> fields in the ItemSearchRequest document.

<To> <Credential> Required

Identifier for the buying organization. The recommendation is to use the same values the buying company sent in the <From><Credential> fields in the ItemSearchRequest document. Oracle Exchange use the same values the buying company sent in the <From><Credential> fields in the ItemSearchRequest document.

<AuthenticatedKey> Optional

Key that the external site generates that Oracle iProcurement can use for all future search requests. It is recommended that the supplier provide the key for faster performance, since it avoids repeat logins.

<ResultsCount> Optional

Number of search results returned in this response. For example, the <ResultSize> in the item search request may have requested 20 search results. The <ResultsCount> will therefore be 20 as requested, or it might be less than 20—for example, if only 17 search results remain.

<TotalResultsCount> Optional

Total number of results in the external catalog that matched the search request. Since it can take time for a server to calculate total results, this field is optional. The number can also be approximate, such as 100+. Although Oracle iProcurement does not display the total results count, the supplier may provide a number if desired.

<SearchTime> Optional

Time it took for the external server to conduct the search, in seconds.

<SortedBy> Optional

Actual method the external site used to sort the search results. The order attribute can be used to indicate whether the sort is ascending (ASC or asc) or descending (DESC or desc). The value indicates the sort-by method—for example, by Price. The attribute and value are not validated or used by Oracle iProcurement. The supplier could specify any.

<Status> Required

Indicator of whether the search request was successful. The code attribute contains the status code. The value for this field is the error message, if any. The following status codes are recognized by Oracle iProcurement; if the supplier specifies one of these codes, the corresponding error message is displayed to the requester:

Table B–12   ItemSearchResponse Status Codes

Status Code	Description	Message Displayed to Requester
200	Success. Items should be returned in the response only if the error code is 200.	(No message; search results are displayed.)
400	Bad request. This should be returned if the request XML cannot be parsed or some mandatory fields are missing. The exact cause of the failure should be indicated in the error message.	The supplier website is not responding. Try again later. If the problem persists, contact your system administrator.
401	Authentication error. This should be returned if the authentication fails. The exact cause of the failure should be indicated in the error message.	The supplier website is not responding. Try again later. If the problem persists, contact your system administrator.
500	Unexpected error. This should be returned if there is any error on the server when handing the request (for example, the database is not available). The exact cause of the failure should be indicated in the error message.	The supplier website is not responding. Try again later. If the problem persists, contact your system administrator.
501	Sort by criteria is not available. This should be returned if the SortBy field in the search request is not supported by the server.	The supplier is unable to sort the results as you have specified.

Table B–12   ItemSearchResponse Status Codes

Status Code	Description	Message Displayed to Requester
502	Too many search results.	The search criteria you entered will return too many matching items. Narrow your search by entering more keywords.
503	Search language not valid. This should be returned if the SearchLanguage field in the search request is not supported by the server.	The catalog language you have specified is not available for this search. Select a different catalog language and try again.

Internally, these status codes map to one or more Oracle iProcurement error codes that the buying company sees.

<ItemInfo> Required if search results are returned

The lineType attribute currently supports only the following value: goods.

<SupplierPartNum> Required

Supplier part number for the item.

<SupplierCode> Required

Identifier for the supplier. The identifier is not language specific (it does not vary by search language). This identifier is used for mapping the supplier to the value in the EDI Location field in the Supplier Sites window, in Model 1: Transparent Punchout from Oracle iProcurement to Oracle Exchange (XML) only.

<SupplierName> Required

Supplier name. This name is language specific (varies by search language). For example, in a transparent punchout to Oracle Exchange, the Exchange uses the <searchLanguage> in the ItemSearchRequest document to determine which language of the catalog to search and to return results from. The name is used for display purposes only.

<Description> <ShortName> Required

Short description for the item.

<Description> <LongName> Optional

Long description for the item.

<UnitPrice> <Money> Required

Price of the item. The currency attribute in the Money field specifies the currency for the price.

<UOMInfo> <UOMCode> Required

Unit of measure code for the item. The code is not language specific (it does not vary by search language). This code is used for mapping the unit of measure.

<UOMInfo> <UOMName> Optional

Unit of measure name for the item. This name is language specific (varies by search language). For example, in a transparent punchout to Oracle Exchange, the Exchange uses the <searchLanguage> in the ItemSearchRequest document to determine which language of the catalog to search and to return results from. The name is used for display purposes only.

<ManufacturerPartNum> Optional

Manufacturer part number for the item.

<ManufacturerName> Optional

Item manufacturer's name.

<ThumbnailImage> <URL> Optional

Web site address for a thumbnail image of the item. The following illustration shows how the thumbnail images for the items display in the search results.

If you do not specify a thumbnail image, a blank space displays, as shown in the following illustration:

Individual requesters can use their preferences to disable thumbnail images. If so, the text description of the item fills the entire horizontal space, with no thumbnail image.

When search results are displayed in the requester's browser, the browser sends an HTTP request to the supplier's Web server for images. For increased security, each image URL can use the <authenticatedKey>; the supplier's Web server can use the key to perform additional authentication of the requester who is accessing the image if desired. (The supplier sends the <authenticatedKey> to Oracle iProcurement, so that the requester's browser can send the key back to the supplier's server to access the image.) If the authentication is successful, the image is retrieved and returned to the browser.

If you do not have thumbnail images but want to use them, specify the same image URL in both the <Thumbnail> and <Image> URL fields. The buyer can use the profile option POR: Thumbnail Width or POR: Thumbnail Height to create a resized

version of the image for thumbnail purposes. See the image management section of the *Oracle iProcurement Implementation Guide* for more information.

<Image> <URL> Optional

URL for a full image of the item, displayed on the **Item Details** page. The following illustration shows how the image displays on the **Item Details** page:

If you do not specify a full image of the item, no image displays, as shown in the following illustration:

When search results are displayed in the requester's browser, the browser sends an HTTP request to the supplier's Web server for images. For increased security, each image URL can use the <authenticatedKey>; the supplier's Web server can use the key to perform additional authentication of the requester who is accessing the image if desired. (The supplier sends the <authenticatedKey> to Oracle iProcurement, so that the requester's browser can send the key back to the supplier's server to access the image.) If the authentication is successful, the image is retrieved and returned to the browser.

\<CategoryInfo\> \<CategoryCode\> Required

Code for the category. The code is not language specific (it does not vary by search language). This code is used for mapping the category.

\<CategoryInfo\> \<CategoryName\> Optional

Name of the category. This name is language specific (varies by search language). For example, in a transparent punchout to Oracle Exchange, the Exchange uses the \<searchLanguage\> in the ItemSearchRequest document to determine which language of the catalog to search and to return results from. The name is used for display purposes only.

\<CatalogType\> Optional

Indicator of whether the item is from a contract. If the item is from a contract, enter the value CONTRACTED in this field. Otherwise, leave this field blank.

\<BuyerContract\> Optional

Contract number if the item is from a contract. If this contract number matches an approved, effective contract purchase agreement number in Oracle Purchasing, the purchase order that ultimately gets created for the item will get created against that contract. For a full description of the flow, buyers should see the Contract Numbers overview provided in the buyer version of this guide.

\<ItemAttribute\> Optional

Use the ItemAttribute fields to specify additional attributes (also known as descriptors) for the item. The type attribute can be Text, Number, Date, or URL; however, Oracle iProcurement treats the attribute values as Text.

See the XML example ItemSearchResponse earlier in this section for example ItemAttribute fields. The illustrations of the **Item Details** page above show how the ItemAttribute fields in this XML example display to the requester. ItemAttribute fields display only on the **Item Details** page. They display at the end, after the other item information.

The ItemAttribute fields can be specific to the search language (the language sent in the \<searchLanguage\> field in the ItemSearchRequest document). For example, the requester's session language is English. If the supplier returns item information in German, the item information itself displays in German; however, the field names, such as item Description or Manufacturer, display in English. The ItemAttribute fields, however, display exactly how they are sent. If the supplier returns both an

ItemAttribute field and its value in German, both the field and its value display in German, regardless of the requester's session language.

When conducting a transparent punchout to Oracle Exchange, any attributes that the Exchange Operator defines, in addition to those such as Price or Manufacturer that Oracle Exchange already provides, are returned in the ItemAttribute fields.

Mapping Between XML and Oracle iProcurement Fields

The table below shows the mapping between XML fields and the fields that requesters see in the search results or item details in Oracle iProcurement.

ItemAttribute fields in an ItemSearchResponse document display in addition to the fields below. For example, Lead Time is a standard item field that Oracle iProcurement provides, but it is not a standard XML field in the ItemSearchResponse document. If the supplier specifies Lead Time in the ItemAttribute field in the ItemSearchResponse document, then that Lead Time displays in addition to the Lead Time field that Oracle iProcurement already provides.

Table B–13   Mapping of XML and cXML Fields to Oracle iProcurement Fields

Field in Oracle iProcurement	XML Field in shoppingCart	cXML Field in PunchoutOrderMessage	XML Field in ItemSearchResponse
Category	<categoryCode>	<Classification>	<CategoryName>
Description	<itemDescription>	<Description>	<ShortName>
Unit	<supplierUOMType>	<UnitOfMeasure>	<UOMName>
Unit Price	<unitPrice>	<Money>	<Money currency=>
Currency	<currency>	<currency>	<Money currency=>
Supplier	<supplierName>	*(Supplier name entered in punchout setup definition)*	<SupplierName>
Supplier Item	<supplierItemNumber> <itemID>	<SupplierPartID>	<SupplierPartNum>
Manufacturer	<manufacturerName>	<ManufacturerName>	<ManufacturerName>
Manufacturer Item	<manufacturerItem Number><itemID>	<ManufacturerPartID>	<ManufacturerPartNum>
Image	*(no XML field)*	*(no XML field)*	<Image><URL>
Image URL	*(no XML field)*	*(no XML field)*	<Image><URL>

Table B–13   Mapping of XML and cXML Fields to Oracle iProcurement Fields

Field in Oracle iProcurement	XML Field in shoppingCart	cXML Field in PunchoutOrderMessage	XML Field in ItemSearchResponse
Thumbnail Image	*(no XML field)*	*(no XML field)*	<ThumbnailImage> <URL>
Long Description	*(no XML field)*	*(no XML field)*	<LongName>
UNSPSC Code	*(Supplier can provide UNSPSC code in <categoryCode>, but it does not display in UNSPSC field in iProcurement.)*	*(Supplier can provide UNSPSC code in <Classification>, but it does not display in UNSPSC field in iProcurement.)*	*(Supplier can provide UNSPSC code in <CategoryName>, but it does not display in UNSPSC field in iProcurement.)*
Functional Currency Price	*(Oracle iProcurement populates this field)*	*(Oracle iProcurement populates this field)*	*(Oracle iProcurement populates this field)*
Item Type	<item lineType=>	*(no XML field)*	<ItemInfo lineType=>
Supplier Site	<supplierSite>	*(no XML field)*	*(no specific XML field)*
Contract Number	<buyerContract> <contractNumber>	*(no XML field)*	<BuyerContract>
Supplier Config ID (hidden field)	<supplierReference Number>	<SupplierPartAuxiliaryID>	*(no specific XML field)*

> **Note:**   Do not confuse the <SupplierPartAuxiliaryID> field in a cXML document with the Supplier Part Auxiliary ID *descriptor* in the local catalog in Oracle iProcurement. These fields are used for different purposes. The <SupplierPartAuxiliaryID> cXML field is used in the cXML document to specify a configuration number. The Supplier Part Auxiliary ID descriptor in the local catalog is used by the bulk loader to specify augmentative item information. For details, see the *Oracle iProcurement Implementation Guide* or the downloadable bulk load instructions in the eContent Manager.

Mapping Between XML and cXML

When Oracle Exchange converts a supplier's cXML shopping cart to XML in a punchout, it converts the fields as described in the table below. (These fields represent values that come from the cXML document itself; other values may come from Oracle Exchange, for example.)

Table B–14   cXML to XML Conversion of Shopping Cart in Punchout

cXML	XML
quantity	<item quantity=…>
SupplierPartID	<supplierItemNumber> <itemID>
SupplierPartAuxiliaryID	<supplierReferenceNumber>
Money	<unitPrice>
currency	<currency>
Description	<itemDescription>
UnitOfMeasure	<supplierUnitOfMeasure> <supplierUOMType>
Classification	<categoryCode>
ManufacturerPartID	<manufacturerItemNumber> <itemID>
ManufacturerName	<manufacturerName>

C

Authentication, Security, and Encoding

This chapter covers the following topics:

Authentication and Security

Oracle iProcurement and Oracle Exchange come with the capability to access secure sites. If the site the buying company accesses is secure (the site URL starts with a secure protocol such as https:// instead of http://), then the buying company should review the prerequisites described in the buyer setup.

> **Note:** Oracle iProcurement and Oracle Exchange do not support client-side certificate authentication. That is, they authenticate the server they are accessing, but do not allow that server to authenticate them in return (known as client-side authentication). If the supplier's server attempts client-side authentication, the connection will fail.

The discussions below use a punchout from Oracle iProcurement to the secure Oracle Exchange site as an example, but the information applies to all models where the site being accessed is secure.

Secure Sockets Layer (SSL) Authentication

Oracle iProcurement uses the SSL protocol to establish a secure HTTP connection between the Oracle iProcurement server and the supplier site or Oracle Exchange. The SSL protocol is a set of rules governing authentication and

encrypted communication between servers and clients. Oracle iProcurement and Oracle Exchange call Oracle SSL application programmable interfaces (APIs) to establish a connection to the secure site.

For example, in a punchout from Oracle iProcurement to Oracle Exchange, the Oracle Exchange server responds to Oracle iProcurement, through the SSL connection, with a digital certificate. Oracle iProcurement then authenticates the digital certificate. A digital certificate is proof that a site (Oracle Exchange in this example) is who it says it is. Established and trusted companies or services known as *certification authorities* assign digital certificates to sites who apply for them. To authenticate the digital certificate sent from Oracle Exchange, Oracle iProcurement compares the digital certificate to certification authorities stored in the ca-bundle.crt file in Oracle iProcurement.

The steps involved in validating the digital certificate are as follows:

- Check the validity period. For example, in a punchout from Oracle iProcurement to Oracle Exchange, the Oracle iProcurement server checks the validity period on the certificate presented by the Oracle Exchange server. If the current date and time is outside the validity period's date range, the authentication process stops.

- Check the certification authority. In Oracle iProcurement, a list of trusted certification authorities is maintained on the server in a file called ca-bundle.crt, which is part of the standard Oracle Application Server installation. If the name of the certification authority on the certificate that the supplier site sends matches the name of a certification authority in the ca-bundle.crt file, the authentication proceeds. Oracle iProcurement and Oracle Exchange also support a certification chain. In a certification chain, the supplier site's certification authority may not match the certification authority in Oracle iProcurement or Oracle Exchange, but it references another certification authority that does match.

- Check that the certification authority validates the digital certificate. For example, in a punchout from Oracle iProcurement to Oracle Exchange, the Oracle iProcurement server uses the certification authority's public key (a code that decrypts certificates) to validate the digital signature on the digital certificate sent by the Oracle Exchange server. The public key is included in the ca-bundle.crt file. Oracle iProcurement and Oracle Exchange use standard techniques of encrypting and decrypting data using public and private keys.

- Check that the domain name on the digital certificate matches the domain name of the sending server itself. For example, in a punchout from Oracle

iProcurement to Oracle Exchange, the Oracle iProcurement server ensures that the Oracle Exchange server is on the same network as that stated in the certificate and is not one posing as the Oracle Exchange server.

Once the above checks are performed by the accessing application (Oracle iProcurement or Oracle Exchange), the server authentication of the site where the catalog resides is complete. Next, the connected servers use the certificate to generate a session key (to be used only for the duration of that connection between the catalog site and Oracle Exchange or Oracle iProcurement), and Oracle iProcurement or Oracle Exchange transmit this key, encrypted, back to the catalog site using the catalog site server's public key. All subsequent communication between Oracle Exchange or Oracle iProcurement and the catalog site is encrypted.

Oracle iProcurement and Oracle Exchange use SSL version 3.0.

Refer to the following documents for more details on SSL and the public encryption methodologies used in SSL:

- http://developer.netscape.com/docs/manuals/security/sslin/contents.htm
- http://home.netscape.com/security/techbriefs/ssl.html

Shopping Cart Transfer

When the shopping cart is passed to Oracle iProcurement or Oracle Exchange, the external site obtains the URL to which the shopping cart is posted from the <returnURL> field in the login Request document. The external site's server submits the shopping cart to the return URL in Oracle iProcurement via a POST through the requester's browser.

Oracle iProcurement and Oracle Exchange also support SSL connections in a proxy server configuration, if proxy servers are used for outbound connections from the server (for example, the Oracle iProcurement server) to the Internet.

Figure C–1   Accessing the Site via the Firewall
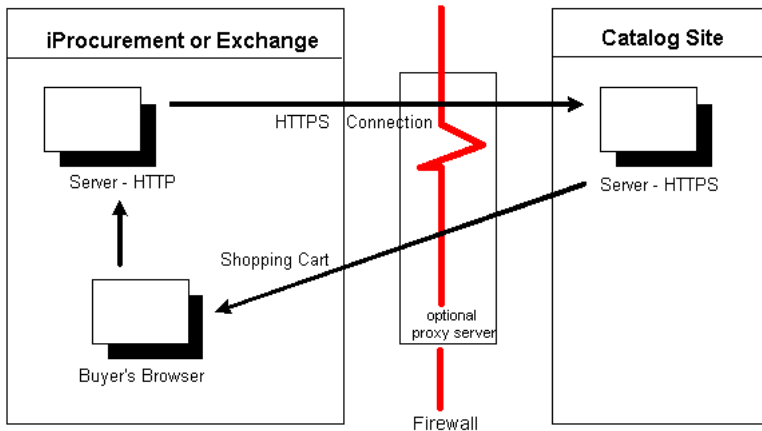
The diagram above shows a non-secure buying site accessing a secure supplier site through the buying organization's firewall (and proxy servers, if any). The shopping cart is returned through the firewall to the browser, and from the browser to the buyer's site. In this diagram, the browser uses HTTP to talk to the buying server and HTTPS to talk to the catalog server.

If the implementation of the Oracle iProcurement server uses HTTPS, then the requester's browser uses HTTPS to connect and submit the shopping cart to the server. All the SSL authentication details discussed above are valid in this process as well.

Encoding

Encoding is specified in the XML prolog as follows:

```
<?xml version="1.0" encoding="UTF8"?>
```

Punchout and transparent punchout use this encoding (not the XML language tag `<language>`) to interpret the XML.

Oracle Exchange supports only UTF-8 character encoding, and Oracle iProcurement works best with punchouts directly to suppliers when UTF-8 encoding is used. Therefore, it is strongly recommended that suppliers always use UTF-8 encoding when sending punchout documents in any model to avoid problems sending and receiving multibyte characters. If the supplier, however,

does use a different encoding, the buying company should specify this encoding on the punchout setup page.

When performing a punchout or transparent punchout to Oracle Exchange, Oracle iProcurement always uses UTF-8 encoding. When performing a punchout from Oracle Exchange, Oracle Exchange always uses UTF-8 encoding. Other encoding methods are not supported by Oracle Exchange.

In a transparent punchout, the XML document is always encoded in UTF-8 so that the supplier's catalog can be searched and results can be returned in any language. For a transparent punchout, the supplier *must* use UTF-8 encoding.

cXML punchouts to a supplier are expected to use UTF-8 encoding.

For XML punchouts, the supplier must URL encode the cart.

D

Maximum Field Lengths

This appendix contains the following sections:

- Punchout from Oracle Exchange on page D-1

- Punchout from Oracle iProcurement on page D-2

- Transparent Punchout from Oracle iProcurement on page D-4

Oracle Exchange and Oracle iProcurement have some data length limits for fields. When the supplier sends the shopping cart to Oracle Exchange or Oracle iProcurement, data greater than the column lengths given in this appendix is truncated at that length. For example, the manufacturer item number (itemID) allows 30 maximum characters. A manufacturer item number of *ABmanufacturer30plusCDEitem123456* sent by the supplier becomes *ABmanufacturer30plusCDEitem123* in Oracle iProcurement or Oracle Exchange.

> **Note:** The lengths given below are *character* lengths. If the data is sent in a multibyte language, the number of *multibyte* characters allowed may be fewer than the character lengths given below.

Punchout from Oracle Exchange

Oracle Exchange has the following maximum character lengths for the shopping cart data when performing a punchout from Oracle Exchange.

Where multiple XML fields are given in the table below, the last field is the one in which the data is contained.

Table D–1   Oracle Exchange Maximum Character Lengths (Punchout)

XML Field	Character Length
<contract> <buyerContractNumber> <contractNumber>	20
<item lineType=Goods quantity=12>	(The quantity is a number.)
<itemNumber> <supplierItemNumber> <itemID>	740
<manufacturerItemNumber> <itemID>	30
<manufacturerName>	255
<itemDescription>	240
<supplierUOMType>	3
<hazardClass>	40
<categoryCode>	250
<currency>	4
<unitPrice>	(This is a number.)
<supplierDUNS>	(This is a number.)
<supplierName>	360
<supplierTradingPartnerCode>	(This is a number.)

Punchout from Oracle iProcurement

Oracle iProcurement has the following maximum character lengths for the shopping cart data.

Where multiple XML fields are given in the table below, the last field is the one in which the data is contained.

Table D–2   Oracle iProcurement Maximum Character Lengths (Punchout)

XML Field	Character Length
<contract>	20
<buyerContractNumber>	
<contractNumber>	
<catalogType>	30
<item lineType=Goods quantity=12>	(The quantity is a number.)
<supplierReferenceNumber>	150
<itemNumber>	25
<supplierItemNumber>	
<itemID>	
<manufacturerItemNumber>	30
<itemID>	
<manufacturerName>	30
<itemDescription>	240
<supplierUOMType>	80
<supplierUOMQuantity>	80
<hazardClass>	(This is a number.)
<categoryCode>	80
<currency>	15
<unitPrice>	(This is a number.)
<supplierDUNS>	35
<supplierName>	240
<supplierTradingPartnerCode>	35
<supplierSite>	240
<ATTRIBUTE1> through <ATTRIBUTE15>	150

Transparent Punchout from Oracle iProcurement

Oracle iProcurement has the following maximum character lengths for item information returned in the search response.

Where multiple XML fields are given in the table below, the last field is the one in which the data is contained.

Table D–3   Oracle iProcurement Maximum Character Lengths (Transparent Punchout)

XML Field	Character Length
<SupplierPartNum>	700
<SupplierCode>	700
<SupplierName>	700
<ShortName>	700
<LongName>	2000
<Money currency=	15
<Money>	(This is a number.)
<UOMCode>	3
<UOMName>	25
<ManufacturerPartNum>	700
<ManufacturerName>	700
<URL>	700
<CategoryCode>	250
<CategoryName>	250
<CatalogType>	10
<BuyerContract>	20

Index

R

S

T

U

V

X

See examples