

Ariba Catalog Format Reference

NOVEMBER 2002



Copyright © 1999-2002 Ariba, Inc. Ariba Proprietary and Confidential. All rights reserved. Patents pending.

Ariba and the Ariba logo are registered trademarks of Ariba, Inc. Ariba Spend Management, Ariba Spend Management. Find it. Get it. Keep it., Ariba Solutions Delivery, Ariba Analysis, Ariba Buyer, Ariba Contracts, Ariba eForms, Ariba Enterprise Sourcing, Ariba Invoice, Ariba Travel & Expense, Ariba Workforce, Ariba Live, Ariba Marketplace, Ariba Network, Ariba Supplier Network, Ariba Network Connect Program, Ariba Network Connectors, Ariba Reach, Ariba SupplierLive, BPM Services, PO-Flip and Ariba Spend Management Knowledge Base, Supplier Advisor, and Making the Net Work for B2B are trademarks or servicemarks of Ariba, Inc.

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

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

Table of Contents

Preface	vii
Audience and Prerequisites	vii
Which Chapters to Read	vii
Related Documentation	viii
Typographic conventions	viii
Ariba Technical Support	ix
Chapter 1	
Introduction to Catalogs	11
Creating and Using Catalogs	11
Catalog Consumers	11
Catalog Creators	12
Ariba Supplier Network	13
PunchOut Catalogs	14
Catalog Formats	15
CIF	15
cXML	15
Deciding Which Format to Use	15
Summary	16
Chapter 2	
Catalog Features	17
Supplier IDs	18
Supplier ID Domains	18
Supplier IDs in Procurement Applications	19
Item Descriptions	19
Item Description	19
Short Name	20
Product Classification Codes	20
UNSPSC	21
Wildcarding	24
Multiple Product Classification Codes	25

Units of Measure	26
PunchOut Catalog Items	27
Display of PunchOut Items	27
PunchOut URL	29
URL vs. SelectedItem	30
Pricing	31
Unit Price and Market Price	31
Price Segments	32
Supplier Part Auxiliary IDs	33
Tiered Pricing	35
Contract Files	37
Territory Available	38
Procurement Applications	39
ISO 3166-1 Country Codes	39
ISO 3166-2 Sublocation Codes	40
Currency	40
Currency Codes	40
Alternate Currency	41
Euro 2002 Conversion	41
Effective Date, Expiration Date, and Lead Time	42
Effective Date	42
Expiration Date	42
Lead Time	42
Supplier and Manufacturer URLs	43
Supplemental Attributes	44
Approved Attributes	44
Attribute Format	45
Parametric Search Data	45
Overview of Parametric Searching	45
Parametric Attribute Names and Values	48
Parametric Name	49
Recommendations for Parametric Search Data	50
Parametric Data Examples	51
Enabling Incremental Loading	54
Searchable Fields	55
Ariba Buyer	55
Ariba Marketplace, Network Edition (AM-NE)	56
Summary	56

Chapter 3	
Internationalization and Localization	59
Language	59
Locale Codes	59
Creating Multilingual Catalogs	60
Character Set and Character Encoding	64
Character Set	64
Character Byte Expansion	65
Character Encoding	65
Chapter 4	
Type Definition Files	67
Overview of Types	67
Types for Parametric Data	68
Types for Supplemental Attributes	68
Naming Restrictions	68
Ariba Buyer Customization	69
Type Definition File Format	69
Basic Format	69
DOCTYPE	70
TypeProvider	71
Type	71
Type Attributes	73
Parametric Data Examples	76
Example 1	76
Example 2	78
Supplemental Attribute Examples	79
Chapter 5	
CIF Catalog Format	83
Overview	83
CIF Versions	83
File Names	84
Basic Format	84
CIF Header Syntax	84
CIF Body Syntax	85
Decimal Values	87
Hash Tables	87

CIF Index Files	88
Header of an Index File	89
Body of an Index File	91
CIF Contract Files	95
Header of a Contract File	95
Body of a Contract File	96
Examples	97
Minimal Number of Fields	97
Using 12 Basic Fields	97
PunchOut Index Example	98
Multinational Example	98
Parametric Data Example	99
Editing CIF Files	100

Chapter 6

cXML Catalog Format	103
Overview	103
About XML	103
About cXML	103
Types of cXML Catalogs	104
cXML Basics	104
File Names	104
File Headers	105
Basic Syntax	105
Load Mode	106
Index Files	107
Adding Items	108
Deleting Items	109
Contract Files	110
Examples	111
Add an Item	111
Delete Items	112
PunchOut Index Example	113

Index	115
-----------------	-----

Preface

This document describes how to create CIF and cXML catalogs for use by procurement systems.

Audience and Prerequisites

This document is intended for use by suppliers that want to make their product or service content available to buying organizations. This document can also be used by buying organizations producing internal catalogs for their own users.

To create catalogs, you need to know how to use a text editor such as Microsoft® Notepad or Word, or a spreadsheet application such as Microsoft® Excel. To set up automatic catalog generation, you must also know how to write scripts or queries to extract information from your product-information management system or database system.

This document does not describe how to upload catalogs to Ariba Supplier Network (see *Ariba SN Supplier's Guide*) or how to import them into Ariba Buyer (see *Ariba Buyer Catalog Management Guide*).

Which Chapters to Read

- **E-commerce Business Managers**—For an overview of catalogs, read Chapter 1, “Introduction to Catalogs.”
- **Catalog Managers**—To learn all the capabilities of catalogs, read Chapter 2, “Catalog Features.” To see the details of CIF format, read Chapter 5, “CIF Catalog Format.”
- **E-commerce Engineers**—To learn background information that is required to automate catalog content creation, read Chapter 2, “Catalog Features.” Web programmers should also read Chapter 6, “cXML Catalog Format.”

Related Documentation

Ariba documentation related to catalogs includes the following:

Ariba SN Supplier's Guide

Describes how to use Ariba Supplier Network (Ariba SN) to upload, validate, and publish catalogs. This manual also describes how to configure Ariba SN to route purchase orders from your customers. Available on Ariba SN.

cXML User's Guide

Describes how to implement PunchOut catalog Websites and how to receive cXML purchase orders. Available at www.cXML.org.

Ariba Buyer Catalog Management Guide

Describes how load and approve product and service content in the Ariba Buyer procurement application. Available with Ariba Buyer.

Typographic conventions

The following table describes the typographic conventions used in this document:

Typeface or symbol	Meaning	Example
<i><AaBbCc123></i>	Text you need to change is italicized and appears between angle brackets.	http://<server>:<port>/i nspector
AaBbCc123	The names of user interface controls, menus, and menu items.	Choose Edit from the File menu.
AaBbCc123	Files and directory names, parameters, fields in CSV files, command lines, and code examples.	There is one line in ReportMeta.csv for each report in the system.
<i>AaBbCc123</i>	The names of books.	For more information, see the <i>Ariba SN Supplier's Guide</i> .

Ariba Technical Support

For assistance with Ariba applications, technical support is available by phone, email, or over the Web. For information on how to contact Ariba Technical Support, refer to the following Web page:

http://connect.ariba.com/TechSupport_Contacting.htm

Chapter 1

Introduction to Catalogs

Catalogs list products and services offered by suppliers to buying organizations. This chapter provides an overview of the following topics:

- [Creating and Using Catalogs](#)
- [Catalog Formats](#)

Creating and Using Catalogs

Catalogs are text files that convey product and service content to buying organizations. Catalogs describe the products and services you offer and the prices you charge. They are the main communication channel from you to your customers.

Catalog Consumers

Catalogs enable organizations that use procurement systems such as Ariba Buyer or Ariba Marketplace, Network Edition (AM-NE) to see your product and service offerings and buy from you. Procurement applications read your catalogs and store them internally. After a buying organization approves your catalogs, your content is visible to users, who can choose catalog items and add them to purchase requisitions.

You can create catalogs for any product or service, regardless of how it is measured, priced, or delivered.

For each item in a catalog, there is basic required information, and optional additional information that enables advanced catalog features.

The following example is a short catalog from an office-supplies company:

```
CIF_I_V3.0
DATA
942888711, 100, , "Blue Ballpoint Pen", 1213376, 1.95, EA, , , , ,
942888711, 101, , "No. 2 Pencil, Wood", 1213377, 1.50, DZN, , , , ,
942888711, 102, , "Rubber Eraser", 1213472, 0.25, PK, , , , ,
942888711, 103, , "Stapler, Standard", 1237461, 2.95, BX, , , , ,
ENDOFDATA
```

For complete information about the fields in this catalog, see Chapter 5, “CIF Catalog Format”

By making the contents of your catalogs concise, complete, and accurate, you enable customers to order the most applicable products or services. Good catalogs are the key to reducing order errors.

Catalog Creators

Catalogs can be created by either suppliers or buying organizations.

- **Suppliers** create catalogs to make their product and service content available to their customers. Suppliers distribute catalogs through Ariba Supplier Network (Ariba SN), e-mail, or FTP.

Typically, suppliers have a dedicated product-information group that maintains product and pricing details and generates catalogs as needed.

Suppliers might be supplier aggregators, which collect, manage, and distribute product content from many suppliers.

- **Buying organizations** create catalogs to consolidate product content from diverse suppliers and make it available to their users. Buying organizations might create their own catalogs because:
 - They have legacy suppliers that do not provide electronic catalog files.
 - They want product descriptions to contain proprietary information or to have a specific format.
 - They provide their own internal products or services.

Typically, catalogs produced by buying organizations are created by their procurement application catalog administrator.

Ariba Supplier Network

Ariba Supplier Network (Ariba SN) is a Web-based service provided by Ariba, Inc. for connecting suppliers to buying organizations. Ariba SN enables you to publish catalogs to buying organizations. Ariba SN also routes purchase orders from buying organizations to you.

Ariba SN offers the following catalog-management features.

Catalog Validation

When you upload CIF or cXML catalogs to your Ariba SN account, Ariba SN automatically performs syntactic validation. It flags errors by line number, so you can quickly find them and fix them.

Catalog Versioning

Ariba SN can store multiple versions of your catalogs. When you update catalogs by uploading new versions, Ariba SN stores them and allows you to choose any version for publishing.

Catalog and Order Testing

Your Ariba SN test account has a catalog/order tester. This tester displays your catalog contents as a buyer would see it. You can examine your item descriptions and other catalog fields to see how users will see them. You can select catalog items and add them to test purchase orders.

You can submit these test purchase orders to exercise the order routing method you configure on Ariba SN.

Catalog Publishing

You can publish each catalog so it is either public (visible to all buying organizations) or private (visible to specific buying organizations). You might want to make both public and private versions of your catalogs.

- Create **public** catalogs for manufacturers' suggested list prices so that buying organizations can source new products from you.

- Create **private** catalogs for pre-arranged, contract pricing for specific buying organizations.

Buying organizations search for companies, industries, and products to find applicable catalogs. They download them and import them into Ariba Buyer or enable them for AM-NE. Procurement application users search them for needed products and services.

For complete information about Ariba SN, see the *Ariba SN Supplier's Guide*.

PunchOut Catalogs

PunchOut catalogs give you an alternative to static catalog files. PunchOut catalogs are live, interactive catalogs running on your Website.

If you have an e-commerce Website, you can modify it to support PunchOut catalogs. PunchOut catalogs communicate with procurement applications over the Internet by using cXML (commerce eXtensible Markup Language), which supports two-way communication of electronic commerce details.

For PunchOut catalogs, procurement applications display a hyperlink instead of product or pricing details. When users click this hyperlink, their Web browser displays a page from your local Website. Depending on how you implement this page, users can browse product options, specify configurations, and select delivery methods. When users are done manipulating this page, they click a button that returns the order information to the procurement application. The fully configured products and their prices appear within the procurement application on users' purchase requisitions.

Your Website can offer previously agreed-upon contract products and prices.

For information about creating PunchOut catalogs, see the *cXML User's Guide*, available on Ariba SN.

Catalog Formats

Ariba procurement applications support two catalog file formats: CIF and cXML. Both formats are simple text files that you can create with any text editor, word processor, or spreadsheet program.

CIF

CIF (Catalog Interchange Format) is a simple comma-separated list of catalog items and their attributes. Ariba procurement applications support both CIF 2.1 and the newer CIF 3.0 format.

For details about CIF catalogs, see Chapter 5, “CIF Catalog Format”

cXML

cXML (commerce eXtensible Markup Language) format is an XML-based language designed specifically for communication of e-commerce business documents. In cXML, catalog data is described with tags and attributes, using a syntax similar to that of HTML. cXML also supports the communication of purchase orders and PunchOut catalogs.

For details about cXML catalogs, see Chapter 6, “cXML Catalog Format”

Deciding Which Format to Use

CIF and cXML support the same basic set of catalog features. When deciding which catalog format to use, consider the source of your product and service content.

- If you manually create catalogs from a hardcopy catalog or from word-processor files, use CIF format. CIF is the most human-readable format.
- If you use spreadsheets or simple text files to store and maintain product and service content, use CIF format. Most applications make it easy to export content in the comma-separated value (CSV) format that comprises CIF files.
- If you use a Product Information Management (PIM) system or a database, it might be better to use cXML format. cXML is the most flexible format, and it is the easiest to automatically generate and parse, because you can rely on XML DTDs (Document Type Definitions). DTDs precisely define the syntax of cXML.

- If you plan to implement PunchOut catalogs in the future (interactive Websites that provide real-time product content and pricing), consider using cXML. Learning cXML early will make the transition to PunchOut catalogs easier, because PunchOut catalogs use cXML.

Before deciding which format to use, learn the basics of both CIF and cXML.

When updating a catalog's contents, do not change its format; use the same format for all versions of a catalog.

Summary

Catalogs are text files that determine the products and services that are listed within procurement applications.

- Both supplier and buyer organizations can create catalogs.
- Suppliers use Ariba SN to publish and distribute catalogs through the Web.
- PunchOut catalogs are live Web-based catalogs that suppliers host.
- There are two file formats for catalogs: CIF and cXML. CIF is easier to create manually, but cXML is more flexible.

Chapter 2

Catalog Features

CIF and cXML formats have features that you can use to reflect your selling model and complex pricing structures. You can also use these features to differentiate your products and services.

This chapter describes the following catalog features:

- **Supplier IDs**
- **Item Descriptions**
 - Item Description
 - Short Name
- **Product Classification Codes**
- **Units of Measure**
- **PunchOut Catalog Items**
- **Pricing**
 - Unit Price and Market Price
 - Price Segments
 - Supplier Part Auxiliary IDs
 - Tiered Pricing
 - Contract Files
- **Territory Available**
- **Currency**
- **Effective Date, Expiration Date, and Lead Time**
- **Supplier and Manufacturer URLs**
- **Supplemental Attributes**
- **Parametric Search Data**
- **Enabling Incremental Loading**
- **Searchable Fields**
- **Summary**

This chapter discusses catalog features, not syntactic details. For file syntax, see Chapter 5, “CIF Catalog Format” or Chapter 6, “cXML Catalog Format”

Note: Not all features are available in both CIF and cXML formats, and not all features are available in all Ariba products. Any incompatibilities are noted.

Supplier IDs

Supplier IDs uniquely identify suppliers. For each catalog item, you identify the supplier that provides that item.

Most catalogs use D-U-N-S[®] (Data Universal Numbering System) or Ariba Supplier Network ID (NetworkId) numbers for Supplier IDs. You declare which identification system the catalog uses.

Multi-supplier catalogs are not supported in cXML or by Ariba Buyer 8; each catalog should refer to only one supplier. Catalog aggregators should produce one catalog file per supplier.

Supplier ID Domains

The *Supplier ID Domain* is the system you use to identify suppliers in your catalogs. You can use any identification system, such as D-U-N-S, NetworkId, or customer-specific Supplier IDs. For easiest integration with procurement applications, use the same Supplier ID Domain that Ariba SN displays when it lists your account information.

The CIF and cXML formats specify Supplier ID Domains differently.

CIF catalogs specify Supplier ID Domain in the file header, and all items in that file use that domain. The following example shows the header for a CIF file that uses D-U-N-S numbers:

```
CIF_I_V3: 0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
```

In cXML catalogs, each item can have multiple Supplier IDs, each in its own domain:

```
<SupplierID domain="DUNS">100220194</SupplierID>
<SupplierID domain="NetworkId">AN000012345</SupplierID>
<SupplierID domain="psoft">XYZ124</SupplierID>
```

Supplier IDs in Procurement Applications

Procurement applications each maintain a list of known suppliers. During catalog import, they match catalog items with a supplier; any items from unknown suppliers are rejected.

Alternatively, they can query Ariba SN for the profiles and IDs of new suppliers.

If you distribute your catalogs through Ariba SN, the supplier list within procurement applications will contain your own Supplier ID. However, if your catalogs contain multiple Supplier IDs, Ariba SN might not have profiles for those suppliers, so procurement application administrators would need to manually enter those suppliers. Similarly, if you send catalogs through e-mail or FTP, procurement application administrators would need to manually enter your supplier information.

Item Descriptions

Each catalog item must have an *Item Description* and can optionally have a *Short Name*. Procurement applications display these fields to help users decide which products and services to purchase.

These fields are case-sensitive and can contain any displayable characters. Use initial capitalization for maximum readability. For example:

Blue Ballpoint Pen
High-Pressure Water Pipe
Carpet, Stain Resistant Indoor/Outdoor

Item Description

Item Description can contain up to 240 characters for CIF 2.1 and up to 2000 characters for CIF 3.0 and cXML. Ariba Buyer 7 allows up to 2000 *bytes*.

If you provide only Item Description (without Short Name), procurement applications truncate it to fit product lists; position the name of the item at the beginning of the Item Description so users can see it.

You can provide multiple Descriptions for an item in different languages. For more information, see “[Language](#)” on page 59.

Short Name

Short Name is available in only CIF 3.0 and cXML.

Short Name appears in the product lists presented to users. It can contain up to 50 characters (up to 30 characters is recommended).

Note: AM-NE does not display Short Name.

Do not duplicate information in Short Name and Item Description. Instead, use Short Name to name the product and Item Description to describe product details. For example:

Short Name: SUX-6000 Computer

Item Description: Hexapentium-powered 33-MHz desktop computer. Three 10-MB disk drives, four CD-ROM drives, two Zip drives, 5-Mbit/s Ethernet card, and 4 MB RAM. Includes two blurry monitors and a keyboard with sticky keys. Mouse not included.

Product Classification Codes

Each catalog item must be tagged with a classification code that indicates the kind of item it is. Product classification codes enable procurement applications to display products and services in a hierarchy that makes sense to users.

In addition, buying organizations can create business rules and approval rules that use classification codes to give special treatment to certain kinds of products. For example, Ariba Buyer can be configured to:

- Prevent classes of employees, such as secretaries, from seeing product types that they never need, such as chemicals.
- Require specific organizations to approve the purchase of certain kinds of products. For example, the Information Technology (IT) department might need to approve all requests for computer-related equipment.

Product classification codes are also used by buying organizations' ERP (Enterprise Resource Planning) systems and for enterprise report generation.

You can use any product classification coding system (called a *classification domain*). However, both your catalogs and procurement applications must use the same system, so if you use a non-standard system, you must communicate its details to all buying organizations that use your catalogs. You can include multiple classification coding systems in the same catalog so it can be used by diverse buying organizations.

UNSPSC

The recommended product classification domain is the United Nations Standard Products and Services Code (UNSPSC) system. This standard is a universally accepted coding system that classifies products and services across a broad range of suppliers and industries. Catalogs that use this system can be used by most buying organizations.

The UNSPSC product classification system contains descriptions of more than 12,000 products and services. The coding structure is hierarchical, which combines similar products into standardized product groups. The finer the granularity of the product description, the more digits in the UNSPSC code, up to 8 digits.

UNSPSC Structure

UNSPSC classification codes are eight-digit numbers. Each code consists of four pairs of digits that create a hierarchy of increasingly specific categories, culminating in specific objects. For example, the UNSPSC code for wooden pencils is

44121706

which is composed of the following categories

- 44** Office equipment, accessories, and supplies
 - 12** Office supplies
 - 17** Writing instruments
 - 06** Wooden pencils

For the complete list of codes, see “[Obtaining UNSPSC Numbers](#)” on page 22.

Benefits of UNSPSC

Besides making product catalogs compatible with most procurement applications, UNSPSC can help both buying organizations and suppliers manage product and service information. It is a consistent ontology that provides the following business advantages:

- *Hierarchical*: Describes products and services to a high degree of specificity while also allowing consolidation of items.
- *Detailed*: Descriptions are available at the product level.
- *Expansive*: Covers all conceivable services and products as well as MRO (Maintenance, Repair, and Operations) and production goods.
- *Universal*: Accepted as a standard for procurement analysis, electronic commerce, and ERP transactions. In addition, because each product category has a unique number, its text description can be translated to the language of local users without creating ambiguity.
- *Expandable*: Meets the specific needs of different industries and companies without losing the benefit of a universal standard.

Buying and selling organizations benefit from uniform classification of products and services. The mutual benefits include facilitation of e-commerce and business automation. In addition, consistency and reliability reduce overall cost of participating in e-commerce.

Obtaining UNSPSC Numbers

To obtain UNSPSC numbers, use the UNSPSC list available on the Web.

Two organizations manage the UNSPSC:

Electronic Commerce Code Management Association (ECCMA)

www.eccma.org/unspsc

United Nations Development Programme (UNDP)

www.unspsc.com

You can use these Websites to look up codes for specific items.

If there is no UNSPSC code for one of your products or services, you can contact these organizations and request one. Within a short period, the managing organization will either create a new code or it will recommend an existing number for you to use.

The following examples show how to specify these two classification domains in both CIF and cXML catalogs:

CIF Header	ECCMA UNSPSC CODEFORMAT: UNSPSC UNDP UNSPSC CODEFORMAT: UNDPUNSPSC
cXML ItemDetail	ECCMA UNSPSC <Classification domain="UNSPSC">10101516</Classification> UNDP UNSPSC <Classification domain="UNDPUNSPSC">10101516</Classification>

After you obtain UNSPSC codes, add them to your inventory lists and price lists so they are included in electronic catalogs you generate. If you use automated inventory systems or product-information management systems, create a new field to contain these codes.

UNSPSC Versions

ECCMA periodically releases new versions of the UNSPSC. Later versions cover more products and services, and reorganize existing commodities and assign them new codes.

Catalogs destined for Ariba Buyer 7.1 and earlier and AM-NE must use the UNSPSC version used by the buying organization. Contact your customers to find out the UNSPSC version they use.

Ariba Buyer 8 allow your catalogs to use different UNSPSC versions from the buying organization. It can automatically map codes from one version to another. To enable this automatic mapping, specify the UNSPSC version in your catalogs destined for Ariba Buyer 8.

For example:

CIF Header	<p>Ariba Buyer 7.1 and earlier and AM-NE: CODEFORMAT: UNSPSC</p> <p>Ariba Buyer 8: CODEFORMAT: UNSPSC_V7.0</p>
cXML ItemDetail	<p>Ariba Buyer 7.1 and earlier and AM-NE: <Classification domain="UNSPSC">10101516</Classification></p> <p>Ariba Buyer 8: <Classification domain="UNSPSC_V7.0">10101516</Classification></p>

Ariba Buyer 8 uses this optional version number to map classification codes to the UNSPSC version used by the particular buying organization.

Alternatives to UNSPSC

Instead of using the UNSPSC classification system, you can use other systems such as industry-specific or company-specific ones. However, these alternatives requires you to communicate with your trading partner to agree on the details of the classification system.

Catalogs that use alternative classification systems are not usable by the entire pool of buying organizations on Ariba SN. They are usable only by organizations that use that particular classification system.

The UNSPSC scheme is the most widely used classification system.

Wildcarding

To convey a higher level category or a mix of items, fill unneeded classification code digits with zeros. This shortening of classification codes is called *wildcarding*.

For example, the UNSPSC system has no code for “assorted cut flowers.” Instead, it has categories for specific types of cut flowers, such as lilies:

10161702

- 10** Live plant or animal material
 - 16** Floriculture products
 - 17** Cut flowers
 - 02** Cut lilies

Because there is no explicit code for “assorted cut flowers,” use the code **10161700**, replacing the last digit pair with zeros. This code provides procurement applications enough information to categorize the item correctly. Requisitioners can perform the final selection by reading the descriptions of items in this category.

If you perform wildcarding, replace only the last pair of digits; replacing more digits creates codes that are too general.

Ariba Buyer 8 allows you to perform wildcarding by leaving unneeded classification code digits off, without padding with zeros. In the above example, you could use the code **101617**.

Multiple Product Classification Codes

Buying organizations can standardize on any product classification system. Even different sites within the same company can use different classification systems. Therefore, catalogs that you create for multiple buying organizations might need to contain multiple product classification codes.

You can add multiple classification codes to any item. Buying organizations use the code that is appropriate for them.

Another reason to use multiple classifications codes is to allow the same catalog to be used by both broad buying organizations and narrow buying organizations. For example, a catalog of chemicals could use both a broad classification system (such as UNSPSC), and a narrow classification system (such as a chemical-industry one). The buying organizations that use your catalog can use the system that they find the most useful.

Note: CIF 2.1 format does not support multiple product classification codes per line item.

To add multiple product classification codes to an item, add a table of product classification codes as a collection of domain-value pairs.

The following CIF 3.0 example shows an item with two product classification codes: one ChemClass code and one specific to the ACME buying organization:

```
{ChemCl ass=55667788; ACME=459876; }
```

The following example shows the same item in a cXML file:

```
<Classification domain="ChemCl ass">55667788</Classification>
<Classification domain="ACME">459876</Classification>
```

When catalogs contain multiple classification codes, procurement applications use the first one with a matching code in their internal commodity code lists.

Units of Measure

Every catalog item has an associated unit of measure, which specifies how the item is packaged or measured; for example, *each*, *box*, *kilogram*, or *dozen*. Your catalogs must use a standardized set of units of measure.

The two commonly used standards are United Nations Units of Measure (UNUOM) and American National Standards Institute (ANSI) Units of Measure.

The UNUOM standard is preferred. This standard is also known as the United Nations Center for the Facilitation of Procedure and Practices for Administration, Commerce, and Transport (UN/CEFACT) codes, and as United Nations Trade Data Elements Directory (UNTDDED) common codes. For information on this standard, see www.unece.org/cefact/rec/add2_f.htm

The following table lists frequently used UNUOM codes:

UNUOM Code	Meaning
EA	each
BX	box
DZN	dozen
GRO	gross
KGM	kilogram
LBR	pound
RO	roll

UNUOM Code	Meaning
PR	pair
PK	pack

As with Supplier IDs and product classification codes, the unit of measure standard used in catalog files must match the one used by buying organizations within their procurement applications. Most buying organizations use UNUOM codes.

PunchOut Catalog Items

You can identify catalog items as *PunchOut catalog* items, which means that the information about those items resides on your Website, not within procurement applications. Both CIF 3.0 and cXML support PunchOut catalog items.

For information about creating a PunchOut site, see the *cXML User's Guide* and the *PunchOut Implementation Guide*, both available on Ariba SN.


Punchout catalog items have less information in the catalog file. For example, any pricing information for these items in the catalog file will be ignored and pricing will instead be set directly by your site for each PunchOut session. PunchOut catalog items always include the URL of the supplier's PunchOut Website.

Display of PunchOut Items

PunchOut catalog items do not display in procurement applications with the usual product details and pricing information. Instead, they display a button for initiating a PunchOut session:

Ariba Buyer 6	Shopping cart button
Ariba Buyer 7 and 8	Red asterisk (*) with Buy from Supplier button
AM-NE	Shop button

For example:

	\$0.00USD	Software House International
---	-----------	------------------------------

Ariba Buyer 6

Dell Computers*	<input type="button" value="Buy from Supplier"/>
Supplier: Dell Computers*	Dell Computers
Supplier Part Number: 4317180300	
ETA (days): 0	

Ariba Buyer 7 and 8

Puchout to SHI	<input type="button" value="Shop"/>
Supplier: MSOffice Supplies	Manufacturer: MMM-C
Supp. Part#: AAA	Mfg. Part#: AAA
ETA: 1 day	UNSPSC: 44103107

AM-NE

PunchOut catalog items

For information about those items, users click the button to browse your Website directly.

The following table lists the fields displayed by Ariba Buyer and AM-NE for PunchOut items.

Field Name	Displayed in Ariba Buyer?	Displayed in AM-NE?
Supplier ID		
Supplier Part ID	•	•
Manufacturer Part ID	•	•
Item Description	•	•
SPSC Code		•
Unit Price		
Unit of Measure		•
Lead Time	“0”	•
Manufacturer Name	•	
Supplier URL		
Manufacturer URL		

Field Name	Displayed in Ariba Buyer?	Displayed in AM-NE?
Market Price		
Tier		
Short Name	•	
Language		
Currency		

PunchOut URL

Ariba SN receives PunchOut requests from buyers and routes them to your PunchOut site. The way Ariba SN determines the URL for your site depends on the cXML versions used by buying organizations and your PunchOut site.

There are three places where you can specify the URL of your PunchOut site. In order of preference, they are:

- [URL Specified in Your cXML Profile](#)
- [URL Specified on Ariba SN](#)
- [URL Specified in Your Index Catalog](#)

URL Specified in Your cXML Profile

If you have a cXML 1.1 (or later) site, it should support the cXML Profile transaction and include its PunchOut URL in its profile. You specify your cXML profile URL in the Configuration area of your Ariba SN account.

Ariba SN periodically sends ProfileRequest documents to query your cXML site. Your cXML site responds with a ProfileResponse document listing URLs for all the cXML requests it supports, including PunchOut requests.

- If you do not specify a profile URL on Ariba SN, it uses the method described below ([URL Specified on Ariba SN](#)) to determine your PunchOut URL.
- If you specify a profile URL on Ariba SN, it routes PunchOut requests to the PunchOutSetupRequest URL listed in your ProfileResponse.

This method is recommended over the other two methods, because if you change any of your URLs, you can easily change your cXML profile without having to log on to Ariba SN or upload new index catalogs.

URL Specified on Ariba SN

You can specify the URL of your PunchOut site in the “PunchOut Setup Request” field in the Configuration area of your Ariba SN account:

- If you do not specify a PunchOut URL on Ariba SN, it uses the method described below ([URL Specified in Your Index Catalog](#)) to determine your PunchOut URL.
- If you specify a PunchOut URL on Ariba SN, it routes PunchOut requests to that URL.

This method is recommended over specifying your URL in index catalogs, because if you change your PunchOut site’s URL, you can easily change this parameter on Ariba SN, without having to distribute new index catalogs.

URL Specified in Your Index Catalog

When procurement applications create a cXML `PunchOutSetupRequest` to start a PunchOut session, they insert the URL from your index catalog into the `SupplierSetup` element.

The following supplier index catalog

```
<PunchoutDetail>
  <Description xml:lang="en-US">Desk Chairs</Description>
  <URL>http://www.workchairs.com/punchout.asp</URL>
  <Classification domain="UNSPSC">513603000</Classification>
</PunchoutDetail>
```

results in the following `PunchOutSetupRequest`

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

Ariba SN routes PunchOut requests to this URL if it has no PunchOut URL stored for your site through either of the above methods.

URL vs. SelectedItem

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

- cXML 1.0 and Ariba Buyer 6.1 do not use SelectedItem. So, the URL in your index catalog is the only way to specify the item to punch out for.
- cXML 1.1 and later, Ariba Buyer 7 and later, and AM-NE use SelectedItem to specify the item to punch out for (by including Supplier Part ID and Supplier Part Auxiliary ID). Your PunchOut site can ignore the URL within the PunchOutSetupRequest, so you can use a made up URL within your index catalog.

Pricing

Catalogs contain the prices of items. In the simplest case, there is a price for each item. But pricing is often more complicated, so catalog formats have features that support more flexible pricing models.

The following specialized pricing features are available:

- **Unit Price and Market Price**
- **Price Segments**
- **Supplier Part Auxiliary IDs**
- **Tiered Pricing**
- **Contract Files**

Unit Price and Market Price

Unit Price is a required field that specifies the cost of an item.

Market Price is an optional field in CIF format that specifies the market price, list price, or manufacturer's suggested retail price (MSRP) of an item. Buying organizations can configure their procurement applications to display this field. cXML format does not have Market Price.

Digit Separators

Some countries use a period (.) as the decimal separator in prices, while others use a comma (,). For example:

5.00
7,25

The following table lists the legal digit separators for prices in CIF and cXML:

Format	Decimal Separator	Thousands Separator
CIF	period or comma 5.00 7,25	not allowed 51234
cXML	period 5.00 7.25	comma (optional) 51234 51,234

Spaces are not allowed in numbers. In CIF catalogs, quote the entire price if using a comma; for example, "7,25".

Procurement applications determine how many digits to display for prices. They support at least four digits to the right of the decimal point, but the number of digits displayed to users is determined by each buying organization.

Currency Symbols

Do not include currency symbols, such as \$, ¥, or £.

Procurement applications set the default currency. To specify a particular currency, see “**Currency**” on page 40.

Price Segments

Price Segments specify different prices for an item based on groupings of buyers within a company. For example, a catalog might have a segment for users in North America and one for users in Europe, with each segment having different prices. You can use price segments to account for different shipping costs for different geographic locations.

Note: In CIF format, price segments can appear only in contract files, not in index files. Ariba SN, AM-NE, and Ariba Buyer 8 do not support contract files.

When a user adds an item to a purchase request, Ariba Buyer automatically selects the price segment based on that user’s group. Users for whom the price segment doesn’t apply see the Unit Price.

Before you can use price segments, you must communicate with each buying organization that will use your catalogs to agree upon the names of each segment. They are arbitrary keywords, but usually they are the names of business units or plant locations. Price segments do not need to be geographic regions; they can be any grouping of users that need to receive different prices for the same items.

The following CIF contract example shows how to segment an item into two prices:

```
{Plant12 = 2.85; Plant4= 2.95}
```

Buyers at Plant12 and Plant4 pay different prices for that item.

The following example shows the same information in cXML format:

```
<ItemSegment segmentKey="Plant12">
  <ContractItem>
    <ItemID>
      <SupplierPartID>12345</SupplierPartID>
    </ItemID>
    <UnitPrice><Money currency="USD">2.85</Money></UnitPrice>
  </ContractItem>
</ItemSegment>
<ItemSegment segmentKey="Plant4">
  <ContractItem>
    <ItemID>
      <SupplierPartID>12345</SupplierPartID>
    </ItemID>
    <UnitPrice><Money currency="USD">2.95</Money></UnitPrice>
  </ContractItem>
</ItemSegment>
```

Note: The price segment feature does not internationalize or localize your catalogs. For information about making catalogs available outside of your country, see Chapter 3, “Internationalization and Localization”.

Supplier Part Auxiliary IDs

You must uniquely identify all catalog items to prevent conflicts. Procurement applications identify items by using a combination of their Supplier Part Number and optional Supplier Part Auxiliary ID.

Ariba Buyer 7.1 and earlier and AM-NE use a combination of Supplier Part Number, Supplier Part Auxiliary ID, and Supplier ID.

Uniquely Identifying Items

If your inventory contains multiple items with the same Supplier Part Number, distinguish them with a *Supplier Part Auxiliary ID*.

For example, you might have the following items in your inventory list:

Supplier: Acme
Part number: 123
Description: **Box** of Flour

Supplier: Acme
Part number: 123
Description: **Bag** of Flour

Because procurement applications do not use item descriptions to differentiate items, they interpret the two items above as duplicates. Buying organizations configure Ariba Buyer either to use only the last item or to issue errors during catalog import.

To prevent duplicate items, assign them Supplier Part Auxiliary IDs:

Supplier: Acme
Part number: 123
Description: Box of Flour
Supplier Part Auxiliary ID: Box

Supplier: Acme
Part number: 123
Description: Bag of Flour
Supplier Part Auxiliary ID: Bag

In the above example, both items are considered unique, because they have different Supplier Part Auxiliary IDs.

Use Supplier Part Auxiliary IDs when you have multiple items with the same part number that vary only by their text description. For example, if your inventory lists contain items such as:

Part #	Description	Colors
1234	Pen	red, blue, or green

Use Supplier Part Auxiliary IDs for all three items, because procurement applications do not use their text descriptions for identification. For the above items, you might use the values *red*, *blue*, and *green* as Supplier Part Auxiliary IDs. Procurement applications then display three items in the catalog window.

Other Uses for Supplier Part Auxiliary ID

Other uses for Supplier Part Auxiliary IDs are:

- To uniquely identify services, which typically do not have part numbers.
- To differentiate multiple items that are the same, but have different lead times or delivery options. For example, you might use the values *ground*, *2-day*, and *overnight* as Supplier Part Auxiliary IDs. You could attach a different price to each item.
- To provide catalog items in multiple languages. For example, an item that is available in Italy, France, and Germany could have three Supplier Part Auxiliary IDs, which creates three catalog items. Each catalog item could have a different Description field (in Italian, French, or German.) Similarly, each item could have a different Currency, Unit Price, Lead Time, and Supplier URL.

For the Supplier Part Auxiliary IDs, use Java locale codes, for example *it_IT*, *fr_FR*, and *de_DE*. For a list of these codes, see:

<http://java.sun.com/products/jdk/1.2/docs/guide/international/locale.doc.html>

Buying organizations can configure Ariba Buyer to determine which catalog items to display based on the locations of users.

For more information, see “[Creating Multilingual Catalogs](#)” on page 60.

- To differentiate items that use tiered pricing (see below).

Tiered Pricing

Tiered pricing enables you to offer different catalog items and prices to categories of member organizations within an AM-NE marketplace. Marketplace hosts can divide their member organizations into arbitrary groups, and you can provide items or prices specific to those groups. Marketplace hosts might want different pricing for any business reason, such as the strategic importance of member organizations.

Note: Tiered pricing is supported only by AM-NE.

Tiered pricing is supported only in CIF 3.0 and in cXML.

Marketplace hosts determine the appropriate groups for their member organizations. For example, groups might be:

- Bronze Members – 5% discount
- Gold Members – 10% discount
- Platinum Members – 15% discount

Marketplace hosts name each group (for example, *Bronze*, *Gold*, and *Platinum*) and assign their member organizations to them. Marketplace hosts then inform you about the group names, and you incorporate this information into your CIF 3.0 or cXML catalogs by using the optional `Tier` field. This field makes catalog items visible to specific member organizations.

If you list multiple items with the same Supplier ID and Supplier Part ID, you must uniquely identify them by assigning the tier name to the `Supplier Part Auxiliary ID` field. For more information, see “[Supplier Part Auxiliary IDs](#)” on page 33,

Members of a tier see items tagged for that tier and items that have no tier.

The following CIF 3.0 catalog uses three tiers:

```

CIF_I_V3.0
CODEFORMAT: UNSPSC
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item
Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer
Name, Supplier URL, Manufacturer URL, Market Price, Supplier Part Auxiliary
ID, Tier
CURRENCY: USD
DATA
9590001, 01, 661, Ballpoint Pen, 44121704, 2.95, EA, , , , , Bronze, Bronze
9590001, 01, 661, Ballpoint Pen, 44121704, 1.95, EA, , , , , Platinum, Platinum
9590001, 02, 662, Fountain Pen, 44121703, 4.00, EA, , , , , Bronze, Bronze
9590001, 02, 662, Fountain Pen, 44121703, 3.00, EA, , , , , Platinum, Platinum
9590001, 03, 663, Laser Pointer, 45111601, 20.95, EA, , , , , Gold, Gold
9590001, 04, 664, Wood Pencil, 44121706, 0.75, EA, , , , ,
ENDOFDATA

```

In the above example, Platinum members see lower prices than Bronze members. Only Gold members can see “Laser Pointer.” All members can see “Wood Pencil,” because its tier field is empty. Tier names appear twice for each item, because they are used in the `Supplier Part Auxiliary ID` field to uniquely identify each item.

In cXML catalogs, tiers are declared through extrinsic elements:

```

<IndexItemAdd>
  <ItemID>
    <SupplierPartID>01</SupplierPartID>

```

```

    <SupplierPartAuxiliaryID>Gold</SupplierPartAuxiliaryID>
  </ItemID>
  <ItemDetail>
    <UnitPrice>
      <Money currency="USD">2.95</Money>
    </UnitPrice>
    <Description xml:lang="en">Ballpoint Pen</Description>
    <UnitOfMeasure>EA</UnitOfMeasure>
    <Classification domain="UNSPSC">44121704</Classification>
    <ManufacturerPartID>661</ManufacturerPartID>
    <ManufacturerName>Acme Pens</ManufacturerName>
    <URL>http://www.acme.com</URL>
    <Extrinsic name="Tier">Gold</Extrinsic>
  </ItemDetail>
  <IndexItemDetail>
    <LeadTime>1</LeadTime>
    <ExpirationDate>2002-06-01</ExpirationDate>
    <EffectiveDate>1999-01-01</EffectiveDate>
  </IndexItemDetail>
</IndexItemAdd>

```

Recommendations:

- Marketplace hosts must negotiate levels of discount with each supplier. They must also create tier names and communicate them to each supplier.
- You can use multiple tiers in one catalog, or create multiple catalogs that each use one tier. There is no technical advantage of one method over the other; use the one that is most convenient for you and the marketplace host.
- Avoid changing tiers or their names. Renaming a tier creates significant catalog reloading work (depending on how many catalogs are published to that tier), so names should be well planned, and they should be changed only if absolutely necessary.

Contract Files

Contract prices are prearranged negotiated prices that you offer to specific customers. For example, you might give a customer a 20% discount from your normal list prices in exchange for their agreement to use you as their exclusive supplier for specific items.

Note: Ariba SN, AM-NE, and Ariba Buyer 8 do not support contract files.

Contract pricing allows you to store prices in a separate file from product descriptions. The pricing file is called a *contract* file and the inventory file is called an *index*. Pricing information typically changes more often than inventory lists, so by keeping prices in a separate contract file, you can update pricing quickly and easily.

When you use contract files, contract prices take precedence over prices in index files.

You do not need to use contract files to offer contract pricing. You can combine pre-negotiated prices with product descriptions in catalog index files, without using separate contract files. However, contract files can make it simpler to offer separate pre-negotiated prices to multiple buying organizations—you can create one index file, and create a small contract file per buying organization.

Each item in a contract file must uniquely identify the item (with Supplier ID, Supplier Part Number, and Supplier Part Auxiliary ID, if appropriate) and provide an associated price. The contract file can contain either unit pricing or price segments.

The following example shows a simple CIF contract file (with price segments):

```
CIF_I_V3.0
SEMANTIC: CONTRACT
FIELDNAMES: Supplier ID, Supplier Part ID, Unit Price, Price Segment
DATA
58, ACF-100, 2.85, {PI ant12=2.75; PI ant14=2.70; }
58, ACF-200, 3.45, {PI ant12=3.35; PI ant14=3.40; }
ENDOFDATA
```

Territory Available

To enable buying organizations to display certain catalog items to only specific users based on geography, use the *Territory Available* field. This field lists one or more country or region codes.

CIF example:

```
"US, GB"
```

cXML example:

<TerritoryAvailable>US</TerritoryAvailable>
<TerritoryAvailable>GB</TerritoryAvailable>

Users who reside outside of the countries or regions listed cannot see that item. Catalog items with no Territory Available field are displayed to all users.

Buying organizations each standardize on a specific territory code standard. The recommended ones are ISO 3166-1 and 3166-2.

Note: Do not use these ISO codes to specify currency or language. For more information, see “**Currency**” on page 40 and “**Language**” on page 59.

Procurement Applications

Ariba Buyer and AM-NE ignore the Territory Available field by default. However, buying organizations can write custom filters to enable it in Ariba Buyer. Similarly, marketplace hosts can activate territory filters to enable it in AM-NE.

Contact your customers to find out whether they have configured their procurement applications to use the this field, and which code standard they use.

ISO 3166-1 Country Codes

Buying organizations most often use ISO 3166-1 country or region codes. Use ISO 3166-1 Alpha-2 (two-character) or Alpha-3 (three-character) codes. The Alpha-2 codes are recommended, and they are the ones most often used.

For example:

Alpha-2: US, GB, FR (recommended)

Alpha-3: USA, GBR, FRA

▼ To Look up Alpha-2 codes (recommended):

- 1 Go to www.unetrades.net.
- 2 Click **Repositories and Codes**.
- 3 Click **Country & Currency Codes**.
- 4 Click **Country & Currency**.
- 5 Click **Countries**.

▼ To Look up Alpha-3 codes:

- 1 Go to stiwww.epfl.ch/utlle/i_so_3166.html.

ISO 3166-2 Sublocation Codes

Buying organizations that need finer granularity for territories use ISO 3166-2 country and subdivision codes. These codes list states and provinces.

For example:

Florida: US-FL

Scotland: GB-SCT

Rhône: FR-69

▼ To Look up ISO 3166-2 codes:

- 1 Go to www.unetrades.net.
- 2 Click **Repositories and Codes**.
- 3 Click **Country & Currency Codes**.
- 4 Click **Sublocation & Maintenance**.
- 5 Click **Country, sublocation ISO 3166-2**.

Currency

With cXML and CIF 3.0, you can specify Currency either at the catalog level or the item level. Typically, catalogs use just one currency, which is specified at the catalog level. However, some catalogs specify the default currency at the catalog level and have some individual items that override that default.

If a catalog specifies no currency, it uses the system currency specified within the procurement application.

Currency Codes

Specify currency with three-letter ISO 4217 currency codes, such as:

Japanese yen: JPY
British pound: GBP
US dollar: USD

For a list of these codes, see www.unetrades.net.

Alternate Currency

cXML catalogs can contain *alternate currency* and *alternate amount* attributes. These attributes enable applications to display amounts in a second currency, such as euros. Alternate values are provided for communication with systems that are not capable of converting currencies.

Ariba Buyer is capable of converting currencies, so only one amount is necessary in catalogs. However, buying organizations' ERP systems that receive input from Ariba Buyer might be unable to convert currencies, so they might require alternate values. Ask your customers whether they need alternate values in catalogs.

Euro 2002 Conversion

The following 12 national currencies were replaced by the euro on January 1, 2002:

Austrian schilling	ATS
Belgian franc	BEF
Finnish markka	FIM
French franc	FRF
German mark	DEM
Greek drachma	GRD
Dutch guilder	NLG
Irish punt	IEP
Italian lira	ITL
Luxembourg franc	LUF
Portuguese escudo	PTE
Spanish peseta	ESP

If your catalogs use these currencies, contact your customers to find out their euro conversion strategy. For more information, see www.euro.ecb.int.

Effective Date, Expiration Date, and Lead Time

For each catalog item, you can specify the calendar period during which it is available and how long it takes to deliver.

Specify Effective Date for the start of availability and Expiration Date for the end of availability. Both of these dates must be in the future. If you do not need to specify the start of availability, leave out the Effective Date. Similarly, if you do not need to specify the end of availability, leave out the Expiration Date.

Note: CIF 2.1 does not support Expiration Date or Effective Date.

Effective Date

Procurement applications do not use Effective Date, but if you provide it, it must be a date in the future. This field is useful only to suppliers.

Expiration Date

When an item expires, procurement applications no longer display it to users. However, any in-progress purchase requisitions or purchase orders will continue to list the item. Use the Expiration Date to limit the duration of negotiated pricing. You can also use it for items that will be superseded by newer items in the future. Neither Ariba SN nor procurement applications issue notification when items expire.

Procurement applications hide entire expired line items, not just their prices.

Lead Time

You can include a Lead Time to inform buying organizations how long it takes to receive an item. Lead Time is the number of business days from when you receive the purchase order to when the buying organization receives the item.

Procurement applications do not allow users to set a Need-by date in purchase orders that comes before the Lead Time.

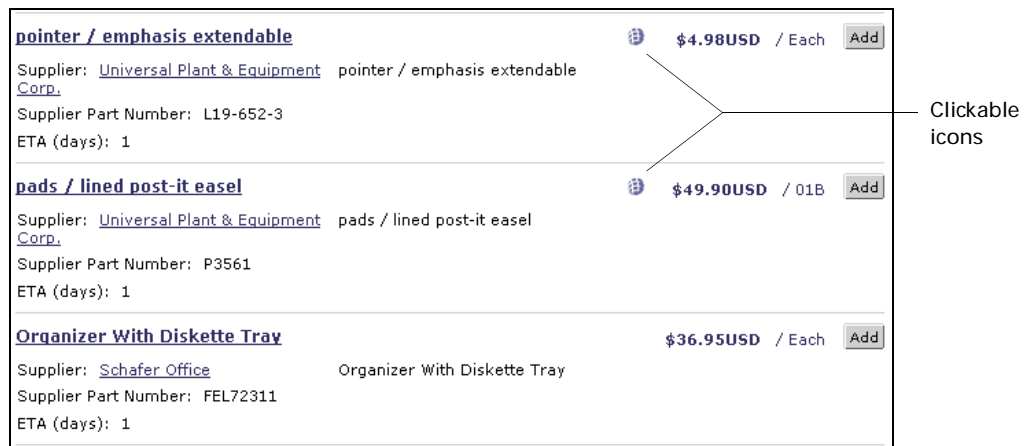
Supplier and Manufacturer URLs

Each catalog item can have a Supplier URL and a Manufacturer URL, which allow users to view information about that item through the World Wide Web.

If either these URLs are present for an item, procurement applications display small clickable Web icons alongside that item. Clicking them displays a new Web browser window containing the specified Website.

The Supplier URL icon displays in the main catalog window. The Manufacturer URL icon displays in the item details page.

The following figure shows Supplier URL icons in a catalog window:



Two items with a Supplier URL

Use these URLs to provide information that helps buyers understand your products or services. For example, you could display:

- Pictures and datasheets for items
- Hazardous material or safety information
- Detailed configuration information about items

- Continuously changing information about items (such as availability or current specifications), which is best maintained on your local Website

These URLs are not used for PunchOut catalogs. None of the information from the supplier or manufacturer Websites is returned to procurement applications for inclusion in purchase orders. To return information from these Websites into procurement applications, use PunchOut catalogs instead.

Note: To access supplier or manufacturer Websites, procurement application users must have Internet access from their desktop computers. Not all buying organizations enable users to access the Internet. Ask your customers whether users have Internet access.

Supplemental Attributes

Supplemental attributes are optional catalog fields that help buying organizations make purchasing decisions. They are not part of the original CIF or cXML catalog format specifications; instead, they convey additional information such as hazardous material or contract ID.

Suppliers can add supplemental attributes to their catalogs, if appropriate. Buying organizations can configure their systems to recognize and take advantage of this additional data.

Ariba Buyer 7.0.6 and later can be configured to recognize and use supplemental attributes. For Ariba Buyer 8, use a *type definition file* to declare supplemental attribute fields. For more information, see Chapter 4, “Type Definition Files”.

Approved Attributes

Ariba SN contains a page that lists the supplemental attributes endorsed by Ariba; Ariba recommends that trading partners use those attributes if additional information must be conveyed in catalogs. This list enables trading partners to use the same standardized attribute keywords so that catalogs are compatible across as many organizations and systems as possible.

The list of endorsed attributes changes over time. For the up-to-date list of attributes and examples of how to use them in CIF and cXML catalogs, log on to Ariba SN.

Attribute Format

Supplemental attribute names must begin with a letter and can include letters, digits, periods (.), dashes (-), and underscores (_). They cannot contain spaces.

The maximum length of a supplemental attribute name combined with its value is 75 characters.

For Ariba Buyer 8 naming restrictions, see “[Naming Restrictions](#)” on page 68.

Parametric Search Data

Ariba Buyer supports optional parametric search data, which enables users to perform sophisticated searching based on product-specific properties. Parametric searching enables users to find items they need within large catalogs of similar products.

This sections discusses the following aspects of parametric search data:

- [Overview of Parametric Searching](#)
- [Parametric Attribute Names and Values](#)
- [Parametric Name](#)
- [Recommendations for Parametric Search Data](#)
- [Parametric Data Examples](#)

Overview of Parametric Searching

To enable parametric searching, add parametric data to items in your CIF 3.0 or cXML catalogs.

For example, if you supply pipes, you could add parametric data to enable users to search by *length* or *diameter*. Or, if you supply computer monitors, you could enable users to search by *screen size* or *dot pitch*.

Note: Ariba Marketplace, Network Edition (AM-NE) does not support parametric searching. It ignores parametric data in catalogs.

Your parametric data causes Ariba Buyer to display new fields in the product search window that all users can use. The following example shows three parametric search properties (SIZE, TYPE, and THREADTYPE) selected in the catalog search page.

Extended Search Parameters for : Casings

SIZE: 10-3/4

THICKNESS:

TYPE: SEAMLESS

THREADTYPE: BUTTRESS THREAD

WEIGHTPERFOOT:

COUPLINGTYPE:

DIAMETERT:

Search Scope:

Catalog Home > Parametric Data > Pipes > Irrigation Pipes

Search Reset

Searching using three parametric search values

Ariba Buyer formulates the contents of this page and its pop-up menus by aggregating the parametric data from your catalogs. You do not need to separately list all parametric data or enumerate all possible parametric attribute values.

You define parametric data as lists of key-value pairs. The following CIF example shows a catalog item with three parametric attributes:

```
{TYPE=SEAMLESS; MATERIAL=STEEL; THICKNESS="10 MM"; }
```

Users see TYPE, MATERIAL, and THICKNESS as search terms. This particular item is returned if a user sets search criteria to TYPE Equals SEAMLESS, MATERIAL Equals STEEL, or THICKNESS Equals 10 MM.

Each catalog item can have only one value per parametric attribute. For example, an item cannot have both MATERIAL=STEEL and MATERIAL=PLASTIC.

CIF Example

The following example shows parametric data in a CIF catalog:

```

CIF_I_V3: 0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item
Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer
Name, Supplier URL, Manufacturer URL, Market Price, Parametric Data,
Parametric Name
TIMESTAMP: 2000-01-18 00:00:00
ITEMCOUNT: 3
DATA
599238,66056062,66056062,"20"" Carbon Steel Seamless Casing",99998050,57.95,E
A,1,CASINGSRUS,,77.95,{SI ZE="20\";"WEIGHT PER FOOT"=94.00;THICKNESS=0.438;THR
EADTYPE="BUTTRESS THREAD";},Casings
599238,66056055,66056055,"20"" Carbon Steel Seamless Casing",99998050,58.95,E
A,1,CASINGSRUS,,78.95,{SI ZE="20\";"WEIGHT PER FOOT"=133.00;THICKNESS=0.635;TH
READTYPE="BUTTRESS THREAD";},Casings
599238,66056048,66056048,"20"" Carbon Steel Seamless Casing",99998050,59.95,E
A,1,CASINGSRUS,,79.95,{SI ZE="20\";"WEIGHT PER FOOT"=137.00;THICKNESS=0.750;TH
READTYPE="BUTTRESS THREAD";},Casings
ENDOFDATA

```

The value 20\" is used because the inch symbol (") must be escaped with a backslash (\). Also, be sure to use a semicolon (;) after each parametric value. For information about specifying parametric data in CIF, see [“Hash Tables”](#) on page 87.

cXML Example

The following example shows parametric data in a cXML catalog.

```

<Item>
  <ItemAdd>
    item ID
    <ItemDetail>
      item description
    </ItemDetail>
    <ItemDetail>
      <SearchGroupData searchGroupName="Casings">
        <SearchDataElement name="TYPE" value="SEAMLESS"/>
        <SearchDataElement name="SI ZE" value=' 20"' />
        <SearchDataElement name="WEIGHT PER FOOT" value=94.00/>
      </SearchGroupData>
    </ItemDetail>
  </ItemAdd>
</Item>

```

cXML type definition and index files are described in Chapter 6, “cXML Catalog Format”.

Declaring Parametric Data Fields

Parametric data fields define the parametric names and search attributes for commodity-specific search data. They define the type of search data, but not the data itself. The search data is part of individual line items in catalogs.

You do not need to declare parametric data fields when using CIF catalogs with Ariba Buyer 7.1 or earlier. You do need to declare them when using cXML, or when using either CIF or cXML with Ariba Buyer 8.

For more information, see “[Parametric Data Examples](#)” on page 51 and Chapter 4, “Type Definition Files”.

Parametric Attribute Names and Values

In CIF, the parametric data type is always *string*. In cXML 1.2.007 and earlier, the parametric data type can be either *string* or *number*, which determines the search modifiers available to users.

cXML Data Type	Search Modifiers Available to Users
string	Equals, Not Equal to
number	Equals, Not Equal to, Greater than, Less than

Beginning with cXML 1.2.008, many data types are allowed for parametric data. For more information, see Chapter 4, “Type Definition Files”.

Case Sensitivity

Parametric attribute names are case-insensitive. Ariba Buyer 8 uses a separate dedicated user-visible value for attribute names.

In Ariba Buyer 7.1 and earlier, parametric attribute values are case-insensitive; it changes them to all uppercase. In Ariba Buyer 8, parametric attribute values are case-insensitive by default, but it can be configured to be case-sensitive.

Requirements

Parametric attribute names and values can use any character set listed in “**Character Set and Character Encoding**” on page 64.

Maximum Length	128 characters (In CIF, each hash table of Parametric Attribute Names with parametric values can contain up to 255 characters.)
First Character	Must be a letter ^a
Legal Characters	Letters, digits, spaces, periods (.), dashes (-), and underscores (_)

a. For the definition of “letter”, see www.w3.org/TR/REC-xml#NT-Letter.

Ariba Buyer removes leading and trailing spaces. Then, it replaces all punctuation (except dashes) with underscores (_).

For Ariba Buyer 8 naming restrictions, see “**Naming Restrictions**” on page 68.

Parametric Name

You assign a *Parametric Name* to your parametric search attributes. Parametric Name (called searchGroupName or Type in cXML) groups attributes to differentiate them from other attributes that have the same name. Each catalog item can have only one Parametric Name.

Parametric Name keeps instances of parametric data from being accidentally combined. For example, if you sell both chairs and pens, and both types of items have the parametric attribute “COLOR,” Parametric Name separates instances of attributes for chairs and pens.

Parametric Name: Chairs	Parametric Name: Pens
COLOR: Black, Brown, Beige	COLOR: Red, Blue, Green
ROLLERS: Yes, No	POINT: Fine, Medium, Course
MATERIAL: Cloth, Vinyl, Leather	

For Parametric Name, use the generic, plural name of the items; for example, Pipes, Chairs, Computers, or Batteries.

Case Sensitivity

Ariba Buyer 6 displays Parametric Name in mixed case. Ariba Buyer 7 and later displays Parametric Name in all uppercase.

Ariba Buyer 8 has a separate dedicated user-visible value for Parametric Name.

Requirements

Parametric Name can use any character set listed in “[Character Set and Character Encoding](#)” on page 64.

Maximum Length	Ariba Buyer 7.0 and earlier: 24 characters Ariba Buyer 7.1 and later: 128 characters
First Character	Must be a letter ^a
Legal Characters	Letters, digits, spaces, periods (.), dashes (-), and underscores (_)

a. For the definition of “letter”, see www.w3.org/TR/REC-xml#NT-Letter.

Ariba Buyer removes leading and trailing spaces. Then, it replaces all punctuation (except dashes) and spaces with underscores (_).

For Ariba Buyer 8 naming restrictions, see “[Naming Restrictions](#)” on page 68.

Parametric Data Aggregation

Ariba Buyer 7.1 and earlier aggregates Parametric Name for all catalogs from a given supplier. Parametric attributes with the same Parametric Name appear aggregated under that Parametric Name. Ariba Buyer 7.1 does not aggregate parametric data from different suppliers.

Ariba Buyer 8 aggregates Parametric Name for all catalogs from all suppliers. This aggregation allows users to search across all suppliers with a single query.

Recommendations for Parametric Search Data

When deciding whether to add parametric search data, consider how users will search for your items. Think about the qualities that differentiate your items from each other.

Also consider the following factors:

- Use parametric data only if you have a large number of similar items, such as variations of a commodity.
- Each catalog item can have only one Parametric Name.
- Keep in mind that you can specify only one value per parametric attribute. For example, an item cannot have both MATERIAL=STEEL and MATERIAL=PLASTIC.
- Do not add parametric data for words that appear in item descriptions. Users can search item descriptions without parametric data.
- Ideal candidates for parametric data are:

COLOR (RED, YELLOW, BLUE)
 MATERIAL (NYLON, POLYETHYLENE, ALUMINUM)
 DIMENSION (1 FOOT, 2 FOOT, 10 FOOT)
 GRADE (A, AA, AAA)
 RATING (12V, 120V, and 240V)

- To keep the search screen manageable, use no more than approximately six different parametric attributes per Parametric Name. For example, a Parametric Name might contain WEIGHT, DIAMETER, COLOR, TYPE, RATING, and MATERIAL.
- To keep search menu length manageable, use no more than approximately 10 different values per parametric attribute. For example, your items might have COLOR attributes of White, Black, Yellow, Green, Blue, Red, Magenta, Brown, and Grey.
- Keep parametric attributes, values, and Parametric Name as short as possible.

Parametric Data Examples

Ariba Buyer 7.1 and earlier (cXML 1.2.007 and earlier) uses SearchGroup elements within catalog files to define parametric data fields.

To learn how to declare parametric data fields in Ariba Buyer 8 (cXML 1.2.008 and later), see Chapter 4, “Type Definition Files”.

Example 1

This example shows parametric data named “Cups” with two attributes named “Material” and “Ounces”.

CIF Catalog

```

CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item
Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer
Name, Supplier URL, Manufacturer URL, Market Price, Parametric Data,
Parametric Name
ITEMCOUNT: 1
DATA
599238,66056,66056,"Large Hot/Cold Drinking Cups",99998050,2.95,DZ,2,Di x i e , , ,
4.95,{MATERIAL="STYROFOAM";OUNCES=12;},CUPS
ENDOFDATA

```

cXML Catalog

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.007/cXML.dtd">
<Index>
  <SupplierID domain="DUNS">942888711</SupplierID>
  <SearchGroup>
    <Name xml:lang="en-US">CUPS</Name>
    <SearchAttribute name="MATERIAL" type="string"/>
    <SearchAttribute name="OUNCES" type="number"/>
  </SearchGroup>
  <IndexItem>
    <IndexItemAdd>
      <ItemID><SupplierPartID>66056</SupplierPartID></ItemID>
      <ItemDetail>
        <UnitPrice><Money currency="USD">2.95</Money></UnitPrice>
        <Description xml:lang="en">
          Large Hot/Cold Drinking Cups
        </Description>
        <UnitOfMeasure>DZ</UnitOfMeasure>
        <Classification domain="UNSPSC">99998050</Classification>
        <ManufacturerPartID>66056</ManufacturerPartID>
        <ManufacturerName>Di x i e</ManufacturerName>
      </ItemDetail>
    </IndexItemAdd>
  </IndexItem>

```

```

        <LeadTime>2</LeadTime>
        <SearchGroupData searchGroupName="CUPS">
            <SearchDataElement name="MATERIAL" value="STYROFOAM"/>
            <SearchDataElement name="OUNCES" value=12/>
        </SearchGroupData>
    </IndexItemDetail>
</IndexItemAdd>
</IndexItem>
</Index>

```

Example 2

This example shows parametric data named “Pipes” with three attributes named TYPE, SIZE, and WEIGHT PER FOOT.

CIF Catalog

CIF header

```

599238, 1234, 1234, "Water Pipe", 99998050, 2.95, FT, 2, Acme, , , 4.95,
{TYPE="SEAMLESS"; SIZE=20\"; "WEIGHT PER FOOT"=94.00; }, PIPES
ENDOFDATA

```

cXML Catalog

header information

```

<Index>
    . . .
    <SearchGroup>
        <Name xml:lang="en-US">Pipes</Name>
        <SearchAttribute name="TYPE" type="string"/>
        <SearchAttribute name="SIZE" type="number"/>
        <SearchAttribute name="WEIGHT PER FOOT" type="number"/>
    </SearchGroup>
</IndexItem>
<IndexItemAdd>
    . . .
    <ItemDetail>
        . . .
    </ItemDetail>
</IndexItemDetail>
    . . .
    <SearchGroupData searchGroupName="Pipes">
        <SearchDataElement name="TYPE" value="SEAMLESS"/>
        <SearchDataElement name="SIZE" value=' 20' />
        <SearchDataElement name="WEIGHT PER FOOT" value=94.00/>
    </SearchGroupData>

```

```

        </IndexItemDetail>
      </IndexItemAdd>
    </IndexItem>
  </Index>

```

Enabling Incremental Loading

When buying organizations receive catalogs, Ariba Buyer 7.1 and earlier can either load the entire files or reload just the items that have changed. Loading all items is called *full loading*; loading only new or changed items is called *incremental loading*. Incremental loading is potentially faster.

Note: AM-NE and Ariba Buyer 8 do not support incremental loading.

Different catalog formats provide different levels of support for incremental loading:

Format	Support for Incremental Loading
CIF	<ul style="list-style-type: none"> • CIF 2.1 offers only full loading. • CIF 3.0 offers full and incremental loading.
cXML	<ul style="list-style-type: none"> • cXML 1.2.007 and earlier offers only incremental loading. • cXML 1.2.008 and later offers full and incremental loading.

Starting with cXML 1.2.008, catalogs can use an optional `loadmode` attribute to indicate whether they are full or incremental. Ariba Buyer 8 and AM-NE support only full catalogs. For more information about the `loadmode` attribute, see “[Load Mode](#)” on page 106.

Use the same file name for all versions of a given catalog. Using the same name makes catalog management easier for buying organizations.

To enable incremental loading, separate individual catalog items into *updates*, *additions*, or *deletions*.

Any item not marked for deletion is either an update or an addition. To determine which are updates, Ariba Buyer compares each item ID, which is the combination of Supplier Name, Supplier Part Number, and Supplier Auxiliary Part ID. The item is an update if its item ID matches one that has already been loaded. Otherwise, the item is an addition.

Explicitly mark items to be deleted. In CIF format there is a Delete column in the file format; set it to TRUE to mark an item for deletion. In cXML, there is an explicit `<itemDelete>` tag that indicates that an item is to be deleted.

Searchable Fields

When users search for products or services, their procurement applications look at specific catalog fields to find them. The catalog fields used depend on the particular procurement application.

Ariba Buyer

Ariba Buyer catalog searching is highly configurable. The default behavior is listed below.

Free text searchable:

- Supplier Part ID
- Parametric Data
- Description
- Parametric Name

Searchable from specific fields, not as free text:

- Price
- Manufacturer Part ID
- Supplier Name
- Lead Time
- Manufacturer Name
- Language

Ariba Buyer ignores all punctuation during searching. It removes accents from characters and any trailing “s” in the search term before searching.

Ariba Buyer administrators can create an equivalent words map, which enables users to perform synonym searching.

Ariba Marketplace, Network Edition (AM-NE)

Free text searchable:

- Supplier ID
- Description
- Supplier Part ID
- Short Name
- Manufacturer Part ID
- Language
- Supplier Part Auxiliary ID
- Classification Code
- Supplier Name
- Punchout Enabled
- Manufacturer Name
- Tier

In addition, AM-NE indexes Supplier Part ID, Manufacturer Part ID, and Supplier Part Auxiliary ID in blocks of four characters, which enables users to search for portions of the ID. For example, KEI76147 can be found by searching for KEI7, EI76, I761, 7614, and 6147. Users can also search for larger blocks (for example, KEI76) and find that item.

Searchable from specific fields, not as free text:

- Price
- Lead Time
- Market Price

Summary

This section is a summary of the features described in this chapter:

- To distinguish multiple items from the same supplier that have the same part number, attach a **Supplier Auxiliary Part ID** to those items. See [page 33](#).
- To maintain catalog data on your Website and allow users to purchase those items from procurement applications, tag those catalog items as **PunchOut catalog items**. See [page 27](#).
- To allow users to search on very specific attributes that apply only to certain catalog items, add **parametric search data** to those items. See “[Parametric Search Data](#)” on page 45.

- To present different groups of users with different prices, associate **price segments** with those items. See [page 32](#).
- To present different prices for variations of an item, create versions of that item using the **Supplier Auxiliary Part ID**. See [page 33](#).
- To publish pricing data separately from item lists, put the pricing data in a separate **contract file**. See [page 37](#).
- To allow **incremental loading**, use cXML format or CIF 3.0 format and explicitly mark items to indicate whether they are additions, changes, or deletions. See [page 54](#).
- To limit the period during which certain catalog items are available, set an **expiration date** for those items. See [page 42](#).
- To enable buying organizations to use a localized configuration, associate **locations** with each catalog item. See [page 38](#).
- To include explicit **currency information**, you can add it at the catalog or the line-item level. See [page 40](#).

Chapter 3

Internationalization and Localization

Both CIF and cXML have optional fields that you can use enable international buying organizations to use your catalogs.

This chapter discusses the following catalog features:

- [Language](#)
- [Character Set and Character Encoding](#)

Language

You can describe catalog items in any language. Both CIF and cXML allow you to specify the name of the language through locale codes, such as `fr_FR` for French and `en_US` for US English. If you do not specify a locale, it is assumed to be the default locale set by the buying organization.

In addition, CIF 3.0 and cXML allow you to provide item descriptions in *multiple* languages. Buying organizations can configure Ariba Buyer to display the appropriate language based on users' locales.

Note: CIF 2.1 does not allow you to specify the language of a description, so you cannot use multiple languages to describe items. Instead, create multiple catalogs, one per language.

Locale Codes

For the names of languages, use Java locale codes; for example, `it_IT`, `fr_FR`, and `en_US`. For a list of Java locale codes, see:

<http://java.sun.com/products/jdk/1.2/docs/guide/international/locale.doc.html>

CIF uses an underbar (`_`) within the codes. cXML uses a dash (`-`); for example *it-IT*, *fr-FR*, and *en-US*.

The country code is needed because of the differences in language from region to region. For example, American English and British English catalogs might contain different spelling.

Creating Multilingual Catalogs

There are several ways to create multilingual catalogs. The method you choose depends on the desires of buying organizations, such as how they want users to be able to view and search catalogs, and what information they want carried forward to requisitions and purchase orders.

You add one line item per language and use the Language field (CIF) or `xml:lang` attribute (cXML) to specify the language of each line item. Differentiate lines that describe the same item by using Supplier Part Auxiliary ID.

Simple Multilingual Catalogs

A buying organization might want you to deliver multilingual catalogs (for example, containing descriptions in both French and English).

- For every SKU item, create two catalog line items – one with description in French and one with description in English.
- There are two additional fields that you must use:
 - **Supplier Part Auxiliary ID** – uniquely identifies items that have the same Supplier ID and Supplier Part ID. Enter “en_EN” to indicate that the description is in English. If this field is used for other purposes, append “en_EN” to the end of the existing text (see example below. It contains UOM and Language.) For more information about this field, see [“Supplier Part Auxiliary IDs”](#) on page 33.
 - **Language** – used by Ariba Buyer to filter for the English or French description products. For example, enter “en_EN” for the English description line item. For more information about this field, see [page 59](#).
- To include prices in the local currency, use the Currency field. (In the example below, the price is in the local currency.)

In addition, the buying organization must add a custom filter to Ariba Buyer so that the appropriate language is displayed. The custom filter should match the user’s locale and the entry in Language field.

Sample CIF 3.0 file segment:

Supplier ID	Supplier Part ID	Item Description	Product Code	Unit Price	Unit of Measure	Supplier Part Aux ID	Language	Currency
0001	01	Blue Pen	1213376	1.95	EA	EA-en_GB	en_GB	GBP
0001	01	Blue Pen	1213376	3.00	DOZ	DOZ-en_GB	en_GB	GBP
0001	01	Stylo Bleu	1213376	1.20	EA	EA-fr_FR	fr_FR	FRF
0001	01	Stylo Bleu	1213376	2.75	DOZ	DOZ-fr_FR	fr_FR	FRF

Advantages and Disadvantages:

- Searching – users can search using their local language.
- Displaying – only the local language description displays.
- Carrying through to requisition – only the local language description appears on requisitions and purchase orders.

Multilingual Catalogs with Territory Available

Buying organizations might want you to deliver multilingual catalogs (for example, containing descriptions in both French and English). However, an organization might have both English-speaking and French-speaking users in its French office. English users must be able to search in English but order items using the French franc pricing.

- For every SKU item, create two catalog line items—one with description in French and one with description in English.
- Use three additional fields to differentiate these items:
 - **Supplier Part Auxiliary ID** – uniquely identifies items that have the same Supplier ID and Supplier Part ID. Enter “en-EN” to indicate that the description is in English. If this field is used for other purposes, append “en-EN” to the end of the existing text (the example below contains both UOM and Language.) For more information about this field, see “**Supplier Part Auxiliary IDs**” on page 33.
 - **Language** – used by Ariba Buyer to filter for the English or French description products. For example, enter “en-EN” for the English description line item.
 - **Territory Available** – Lists locations where the product is available for purchase. For more information about this field, see “**Territory Available**” on page 38.

- Use the Currency field to include prices in the local currency. (In the next example, the price is in the local currency.)

The buying organization must add custom filters to Ariba Buyer to display the appropriate language and allow users to search only for items available in their territory. The custom filters should match users' locales for both the Language field and the Territory Available field.

Sample CIF 3.0 file segment:

Supplier ID	Supplier Part ID	Item Description	Product Code	Unit Price	Unit of Measure	Supplier Part Aux ID	Language	Territory Available	Currency
0001	01	Blue Pen	1213376	1.95	EA	EA-en_GB	en_GB	GB	GBP
0001	01	Blue Pen	1213376	3.00	DOZ	DOZ-en_GB	en_GB	GB	GBP
0001	01	Blue Pen	1213376	1.20	EA	EA-en_GB	en_GB	FR	FRF
0001	01	Blue Pen	1213376	2.75	DOZ	DOZ-en_GB	en_GB	FR	FRF
0001	01	Stylo Bleu	1213376	1.20	EA	EA-fr_FR	fr_FR	FR	FRF
0001	01	Stylo Bleu	1213376	2.75	DOZ	DOZ-fr_FR	fr_FR	FR	FRF

Advantages and Disadvantages:

- Searching – users can search using only their local language.
- Displaying – only the local language description displays.
- Carrying through to requisition – only the local language description appears on requisitions and purchase orders.

Multilingual Catalogs Combining Languages on One Line

A buying organization might want you to supply catalogs that allow both English and French descriptions to display to users at the same time. There are two possible solutions: using the Manufacturer URL field, or using multiple languages in the Description field.

- Solution 1: Manufacturer URL

You can enter the English description in the Item Description field and enter the French description in the Manufacturer URL (or another field). The Manufacturer URL field is suggested because it can contain up to 255 characters; however, other unused fields can be substituted.

Sample CIF 3.0 file segment:

Supplier ID	Supplier Part ID	Item Description	Product Code	Unit Price	Unit of Measure	Supplier Part Aux ID	Manufacturer URL	Currency
0001	01	Blue Pen	1213376	1.95	EA	EA	Stylo Bleu	GBP
0001	01	Blue Pen	1213376	3.00	DOZ	DOZ	Stylo Bleu	GBP

Advantages and Disadvantages:

- Searching – users can search on both English and French descriptions. However, Ariba Buyer would have to be customized to search against the Manufacturer URL field.
- Displaying – both languages display.
- Carrying through to requisition – Ariba Buyer must be configured to include Manufacturer URL as an extrinsic on requisitions and purchase orders because it does not include that field by default.
- Solution 2: Multiple Languages in Description

You can enter both English and French descriptions into a single Description field, separated by punctuation, such as a slash (for example, “PEN/STYLO”).

Sample CIF 3.0 file segment:

Supplier ID	Supplier Part ID	Item Description	Product Code	Unit Price	Unit of Measure	Supplier Part Aux ID	Currency
0001	01	Blue Pen / Stylo Bleu	1213376	1.95	EA	EA	GBP
0001	01	Blue Pen / Stylo Bleu	1213376	3.00	DOZ	DOZ	GBP

Advantages and Disadvantages:

- Searching – users can search on both English and French descriptions, without additional customization of Ariba Buyer.
- Displaying – both languages will display.
- Carrying through to requisition – both language descriptions can appear on requisitions and purchase orders. The Description field accommodates 2000 characters (2000 bytes in Ariba Buyer R7), but only the first 255 characters appear on requisitions and purchase orders.

Character Set and Character Encoding

When creating catalogs, you should understand the implications of the character set and character encoding you use.

Character Set

Catalogs can specify the character set they use. If they specify no character set, Ariba procurement applications use the Cp1252 (Windows Latin-1) character set to display catalog contents.

Note: CIF 2.1 does not allow you to specify the character set; it uses Cp1252 (Windows Latin-1).

In CIF 3.0, use the CHARSET header field to specify the character set. In cXML, use the `?xml encoding` attribute to specify the character set. In either format, you can use UTF-8 (Unicode), which is a universal encoding that supports all character sets.

The following table lists the character sets supported in catalogs by Ariba applications:

Description	Code ^a	Supported by					
		Ariba SN	Ariba Buyer				AM-NE
			R6.1	R7.0.5	R7.0.6	R7.1 and 8.0	
Western European languages	8859_1	•	•	•	•	•	•
	Cp1252	•	•	•	•	•	•
Japanese ^b	Shift_JIS	•		•	•	•	
	MS932	•		•	•	•	
Simplified Chinese	GB2312	•		•	•		
	EUC_CN	•		•	•		
	MS936	•		•	•		
Traditional Chinese	Big5	•		•	•		
	EUC_TW	•		•	•		
	MS950	•		•	•		
Korean	KSC5601	•		•			
	EUC_KR	•		•			
	MS950	•		•			

- a. The code UTF-8 includes all the above character sets
- b. Suppliers providing Japanese content should use Zenkaku Katakana characters, not Hankaku Katakana characters. Hankaku Katakana uses more space in procurement application databases and can cause display problems in some Web browsers and email readers. Additionally, if it appears in part IDs, applications cannot search for those IDs correctly.

Character Byte Expansion

When buying organizations load catalogs into their procurement applications, UTF-8 non-ASCII characters expand to a number of bytes larger than the number of original characters. Extended-ASCII characters expand to two bytes, and Asian characters expand to three bytes.

For example, 10 characters of full-width Japanese expand to 30 bytes within procurement applications. Considering the 2000-byte limitation of item descriptions, Japanese item descriptions can have at most 666 characters.

European languages have a mix of US-ASCII and extended-ASCII, so the actual maximum number of characters per catalog field depends on the ratio of that mix.

Character Encoding

Multilingual catalogs that contain otherwise incompatible characters (for example, French and Japanese) must be saved in UTF-8 format. In addition, the CIF 3.0 CHARSET value or cXML encoding attribute must be set to UTF-8.

Raw Unicode, also called UCS-2 (Universal Character Set encoded in two octets), is not file-safe because its byte order varies depending on the operating system. However, UTF-8 (UCS Transformation Format, 8-bit form) is a file-safe encoding of that character set.

If you use Windows Notepad to edit catalogs, do not click the “Save as Unicode” checkbox, because that option saves in raw Unicode format. For the same reason, if you use Microsoft Word, do not select “Unicode Text” as the file type.

If you e-mail your catalogs, ensure that the encoding of the e-mail matches the encoding of the catalog, such as UTF-8.

Chapter 4

Type Definition Files

Ariba Buyer 8 uses dedicated cXML files called *type definition files* to define parametric search attributes and supplemental catalog attributes.

This sections discusses the following aspects of type definition files:

- [Overview of Types](#)
- [Type Definition File Format](#)
- [Parametric Data Examples](#)
- [Supplemental Attribute Examples](#)

Overview of Types

Type definition files extend the catalog format used by Ariba Buyer 8. They define *types*, which are elements that create custom catalog fields with specific attributes. Custom catalog fields enable suppliers to add parametric data or supplemental attributes to their catalogs.

Ariba Buyer uses two type definition files: one for parametric data types and one for supplemental attributes. These files are required by Ariba Buyer 8 regardless of whether parametric data or supplemental attribute data comes from CIF or cXML catalogs.

These files can be created by buying organizations, suppliers, or supplier aggregators. Typically, they are created by catalog managers within buying organizations. Catalog managers then communicate their catalog data requirements so their suppliers know the exact field names to use in catalogs.

This section provides an overview of types for Ariba Buyer parametric data types and supplemental attributes.

Types for Parametric Data

Types define the Parametric Names and attributes for commodity-specific search data. They define the type of search data, but not the data itself. The search data is part of individual line items in catalogs. For an overview of parametric data, see [“Parametric Search Data”](#) on page 45.

Note: To declare parametric data fields for Ariba Buyer 7.1 or earlier, see [“Parametric Search Data”](#) on page 45.

Ariba Buyer 8 ignores SearchGroup elements in cXML catalogs. SearchGroup elements are used only by Ariba Buyer 7.1 (cXML 1.2.007) and earlier.

Types for Supplemental Attributes

Supplemental attributes are custom catalog fields agreed upon by a buying organization and its suppliers for specific business purposes. They are not part of the basic CIF or cXML catalog format. For an overview of supplemental attributes, see [“Supplemental Attributes”](#) on page 44.

Naming Restrictions

Parametric and supplemental attribute names have the regular restrictions described in [“Parametric Attribute Names and Values”](#) on page 48 and [“Supplemental Attributes”](#) on page 44.

In addition, Ariba Buyer 8 requires supplemental and parametric attribute names to be different than the attribute names of the base CatalogItem type:

TypeName
SupplierId
SupplierName
SupplierPartId
SupplierPartAuxiliaryId
ManufacturerPartId
ManufacturerName
ManufacturerURL
URL
Description
ShortName
UnitOfMeasure
NonContractPrice
ContractPrice

Price
ExpirationDate
EffectiveDate
LeadTime
Language
ClassificationCode
TerritoryAvailable
PunchoutEnabled
Key

Do not use these values for any supplemental or parametric attribute names.

Ariba Buyer Customization

By default, Ariba Buyer does not recognize and use parametric data or supplemental attributes from catalogs. Buying organizations must customize Ariba Buyer to use these fields.

For instructions on customizing Ariba Buyer to use these fields, see the *Ariba Buyer Catalog Management Guide*.

Type Definition File Format

A type definition file is a cXML TypeDefinition document that defines either parametric data types or supplemental catalog attributes.

This section discusses type definition files in terms of:

- **Basic Format**
- **DOCTYPE**
- **TypeProvider**
- **Type**
- **Type Attributes**

Basic Format

TypeDefinition documents contain the following elements:

TypeProvider — identifies the organization creating the types.

Type — specifies the names and other attributes of custom catalog fields.

The syntax of TypeDefini ti on documents is described by the cXML Document Type Definition (DTD) Catal og. dtd, available at [www. cXML. org](http://www.cXML.org).

The following type definition file defines a parametric data type name of “Cups” with two parametric data attributes named “Material” and “Ounces”:

```
<?xml versi on="1.0" encodi ng="UTF-8"?>
<!DOCTYPE TypeDefini ti on SYSTEM
"http://xml . cxml . org/schemas/cXML/1. 2. 008/Catal og. dtd">
<TypeDefini ti on>
  <TypeProvi der name="System">
    <Organi zati onID>
      <Credenti al domai n="buyersystemi d">
        <Identi ty>1</Identi ty>
      </Credenti al >
    </Organi zati onID>
  </TypeProvi der>
  <Type name="CUPS" extends="System: Catal ogI tem">
    <Descri pti on xml : l ang="en">Standard Dri nki ng Cups</Descri pti on>
    <Descri pti on xml : l ang="de">Tri nkende Standardschal en</Descri pti on>
    <Name xml : l ang="en">Cups</Name>
    <Name xml : l ang="de">Schal en</Name>
    <TypeAttri bute name="MATERI AL"
      type="Name"
      i sRefi nabl e="yes">
      <Name xml : l ang="en">Materi al </Name>
      <Name xml : l ang="de">Materi al </Name>
    </TypeAttri bute>
    <TypeAttri bute name="OUNCES"
      type="Number"
      i sRefi nabl e="yes">
      <Name xml : l ang="en">Ounces</Name>
      <Name xml : l ang="de">Unzen</Name>
    </TypeAttri bute>
  </Type>
</TypeDefini ti on>
```

The major elements in this file are described below.

DOCTYPE

DOCTYPE is a standard XML document type declaration. It must refer to the DTD Catal og. dtd from cXML 1.2.008 or later.

TypeProvider

The cXML standard allows any value in the `TypeProvider` name attribute. However, Ariba Buyer accepts only the name "System" for this attribute.

Ariba Buyer does not check the type provider's credential.

Type

Type is the basic element within type definition files. It specifies the name of the type and the object it extends.

Use one Type element per Parametric Name. Use one Type element for all supplemental attributes.

Type has the following attributes:

Attribute	Description
name (required)	Specifies the canonical name of the type: <Parametric Name> for parametric data. For more information, see "Parametric Name" on page 49. "CatalogItem" for supplemental attributes.
extends (required)	Specifies the Ariba Buyer object to which to apply the type: "System:CatalogItem" for parametric data. "System:CatalogItemInternal " for supplemental attributes.

Type contains the following elements:

Element	Description
Descripti on	<p>Specifies a description for the type. Create a useful description that differentiates this type from others.</p> <p>This value is visible to catalog managers and Ariba Buyer administrators, not end users. For multilingual configurations, you can specify a name for each locale. Maximum length is 1024 characters.</p>
Name	<p>Specifies the user-visible name for this attribute. For multilingual configurations, you can specify a name for each locale.</p>
TypeAttri bute	<p>Specifies the attributes for this data type. For a description of the properties you can use with this tag, see Type Attributes, below.</p>

Type Attributes

The `TypeAttribute` element defines the characteristics of the data within the type.

`TypeAttribute` has the following attributes:

Attribute	Description
name (required)	Specifies the canonical name of the attribute. For more information, see “ Parametric Attribute Names and Values ” on page 48. Do not use any of the values listed in “ Naming Restrictions ” on page 68.
type (required)	Defines the data type for the attribute. Available data types are: <ul style="list-style-type: none"> "integer" A whole number, with no fraction "string" A group of characters with words that can be individually indexed for free text searching "literal" A group of characters with words that cannot be individually indexed for free text searching "double" A floating point number "date" A date of the form yyyy-mm-dd; for example, 2002-01-25 "boolean" A Boolean value; yes, no, 1, 0, true, false, t, f, T, F, or any mixed lettercase of true, false, yes, or no. "Description" A default primitive in Ariba Buyer, defined as "string" containing up to 2000 characters "Id" A default primitive in Ariba Buyer, defined as "literal" containing up to 128 characters "Name" A default primitive in Ariba Buyer, defined as "string" containing up to 255 characters "Number" A default primitive in Ariba Buyer, defined as "double" "Price" A default primitive in Ariba Buyer, defined as "double" "URL" A default primitive in Ariba Buyer, defined as a "string" containing 5 to 255 characters

Attribute	Description
isCaseSensitive	Set to "yes" to preserve lettercase in the attribute. This property applies only to attributes with type="Literal". It has no effect on other types of attributes.
isCollection	Set to "yes" to allow the attribute to contain repeating entries. For internal Ariba Buyer use; not for use by parametric or supplemental attributes.
isHidden	Set to "yes" to prevent the attribute from displaying for users. Use isHidden when you want to use an attribute for custom filters but do not want it to display in the user interface.
isInFreeTextSearch	Set to "yes" to include the attribute in free text searches performed by users. Set to "no" if this attribute contains values that are not useful in a full-text search, such as single-digit numbers.
isInKey	Set to "yes" to include the attribute as part of the key that Ariba Buyer uses to uniquely identify each catalog item; set to "no" if it is not part of the key. Use for supplemental attributes only.
isReadOnly	Set to "yes" to prevent the attribute from being changed by Ariba Buyer catalog managers or administrators.
isRefinable	Set to "yes" to display the attribute value in a drop-down list of available values; set to "no" to display it as a text box in which users can type the value to search for. You might want to display a text box instead of a list of values (set this attribute to "no") if every item has a different value for this field.
isRequired	Set to "yes" to require a non-empty value for the attribute. If you set this property to "yes", each catalog is deactivated, revalidated, and only reactivated if every catalog item it contains has a value for this field.
isSearchable	Indicates whether the attribute is searchable in search queries. All searchable attributes are indexed. Set to "yes" to make this field searchable; set to "no" to display the field in item details but do not want it available for search.

Attribute	Description
isSortable	Set to "yes" to enable users to sort lists of catalog items by using this field. This property applies only to attributes with type set to a numeric type. It has no effect other types of attributes.
mappedFrom	Specifies the name of an Ariba Buyer internal object that implicitly defines the attribute. The mapping occurs during catalog loading in Ariba Buyer.
unit	Specifies the unit of measure for the attribute data. Not currently used.

The default value for all flags (attributes that start with "is") is "no".

Enumerated Values

TypeAttribute elements can include a list of possible values with EnumerationValue elements.

For example:

```
<TypeAttribute name="MATERIAL"
  type="Name"
  isSortable="yes">
  <Name xml:lang="en">Material</Name>
  <EnumerationValue>Titanium</EnumerationValue>
  <EnumerationValue>Steel</EnumerationValue>
  <EnumerationValue>Aluminum</EnumerationValue>
  <EnumerationValue>Plastic</EnumerationValue>
</TypeAttribute>
```

Ranges

TypeAttribute elements can include a range of possible values with the Range element.

For example:

```
<TypeAttribute name="OUNCES"
  type="Number"
  isSortable="yes">
  <Name xml:lang="en">Ounces</Name>
  <Range>
```

```

    <RangeBegin>8</RangeBegin>
    <RangeEnd>32</RangeEnd>
  </Range>
</TypeAttribute>

```

You can leave off either `RangeBegin` or `RangeEnd` to leave the beginning or ending of the range open.

Parametric Data Examples

The following examples demonstrate how to define parametric data types for Ariba Buyer 8 and how to use them in catalogs.

Catalogs are shown in both CIF and cXML format.

Example 1

This example shows parametric data named “Cups” with two attributes named “Material” and “Ounces”.

Type Definition File

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TypeDefinition SYSTEM
"http://xml.cxml.org/schemas/cXML/1.2.008/Catalog.dtd">
<TypeDefinition>
  <TypeProvider name="System">
    <Organization>
      <Credential domain="buyersystemid">
        <Identity>1</Identity>
      </Credential>
    </Organization>
  </TypeProvider>
  <Comments> Parametric data fields for drinking cups </Comments>
  <Type name="CUPS" extends="System:CatalogItem">
    <Description xml:lang="en">Standard Drinking Cups</Description>
    <Name xml:lang="en">Cups</Name>
    <TypeAttribute name="MATERIAL"
      type="Name">
      <Name xml:lang="en">Material</Name>
    </TypeAttribute>
    <TypeAttribute name="OUNCES"
      type="Number">

```

```

        <Name xml:lang="en">Ounces</Name>
      </TypeAttribute>
    </Type>
  </TypeDefinition>

```

CIF Catalog

```

CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item
Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer
Name, Supplier URL, Manufacturer URL, Market Price, Parametric Data,
Parametric Name
ITEMCOUNT: 1
DATA
599238,66056,66056,"Large Hot/Cold Drinking Cups",99998050,2.95,DZ,2,Di xie,,
4.95,{MATERIAL="STYROFOAM";OUNCES=12;},CUPS
ENDOFDATA

```

cXML Catalog

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index>
  <IndexItem>
    <IndexItemAdd>
      <ItemID><SupplierPartID>66056</SupplierPartID></ItemID>
      <ItemDetail>
        <UnitPrice><Money currency="USD">2.95</Money></UnitPrice>
        <Description xml:lang="en">
          Large Hot/Cold Drinking Cups
        </Description>
        <UnitOfMeasure>DZ</UnitOfMeasure>
        <Classification domain="UNSPSC">99998050</Classification>
        <ManufacturerPartID>66056</ManufacturerPartID>
        <ManufacturerName>Di xie</ManufacturerName>
      </ItemDetail>
    </IndexItemDetail>
    <LeadTime>2</LeadTime>
    <SearchGroupData searchGroupName="CUPS">
      <SearchDataElement name="MATERIAL" value="STYROFOAM"/>
      <SearchDataElement name="OUNCES" value="12"/>
    </SearchGroupData>
  </IndexItemDetail>
</IndexItemAdd>

```

```

    </I ndexI tem>
</I ndex>

```

Example 2

This example shows parametric data named Pipes with three attributes named Type, Size, and Weight per Foot.

Type Definition File

```

header i nformati on
<TypeDefni ti on>
  <Type name="PIPES" extends="System: Catal ogI tem">
    <Descri pti on xml : I ang="en">Pi pes</Descri pti on>
    <Name xml : I ang="en">Pi pes</Name>
    <Name xml : I ang="fr">Tuyaux</Name>
    <TypeAttri bute name="TYPE"
      type="Name"
      i sRefi nabl e="yes">
      <Name xml : I ang="en">Type</Name>
      <Name xml : I ang="fr">Esreçe</Name>
    </TypeAttri bute>
    <TypeAttri bute name="SI ZE"
      type="Number"
      i sRefi nabl e="yes">
      <Name xml : I ang="en">Si ze</Name>
      <Name xml : I ang="fr">Poi nture</Name>
    </TypeAttri bute>
    <TypeAttri bute name="WEI GHT PER FOOT"
      type="Number"
      i sRefi nabl e="yes">
      <Name xml : I ang="en">Wei ght per Foot</Name>
      <Name xml : I ang="fr">Poi ds par Foot</Name>
    </TypeAttri bute>
  </Type>
</TypeDefni ti on>

```

CIF Catalog

```

CIF Header
DATA
599238, 1234, 1234, "Water Pi pe", 99998050, 2. 95, FT, 2, Acme, , , 4. 95,
{TYPE="SEAMLESS"; SI ZE=20\"; "WEI GHT PER FOOT"=94. 00; }, PIPES
ENDOFDATA

```

cXML Catalog

header information

```
<Index>
  <IndexItem>
    <IndexItemAdd>
      . . .
      <ItemDetail>
        . . .
      </ItemDetail>
    <IndexItemDetail>
      . . .
      <SearchGroupData searchGroupName="Pipes">
        <SearchDataElement name="TYPE" value="SEAMLESS"/>
        <SearchDataElement name="SIZE" value="20"/>
        <SearchDataElement name="WEIGHT PER FOOT" value="94.00"/>
      </SearchGroupData>
    </IndexItemDetail>
  </IndexItemAdd>
</IndexItem>
</Index>
```

Supplemental Attribute Examples

The following examples demonstrate how to define supplemental attributes and how to use them in catalogs.

Catalogs are shown in both CIF and cXML format.

Type Definition File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TypeDefinition SYSTEM
"http://xml.cxml.org/schemas/cXML/1.2.008/Catalog.dtd">
<TypeDefinition>
  <Comments> Custom catalog fields for Acme, Inc. </Comments>
  <TypeProvider name="System">
    <Organization>
      <Credential domain="buyersystemid">
        <Identity>1</Identity>
      </Credential>
    </Organization>
  </TypeProvider>
  <Type name="CatalogItem" extends="System:CatalogItemInternal">
    <Name xml:lang="en">CatalogItem</Name>
```

```

<Name xml:lang="fr">Article de Catalogue</Name>
<Description xml:lang="en">Supplemental catalog fields</Description>
<!-- Below is a 12-char field for the source ID of catalog items -->
<TypeAttribute name="Catalog_Item_Source"
  type="Literal"
  min="1"
  max="12"
  isRefinable="yes">
  <Name xml:lang="en">Catalog Item Source</Name>
  <Name xml:lang="fr">Source d'Article de Catalogue</Name>
</TypeAttribute>
<!-- Below is an endorsed supplemental attribute -->
<TypeAttribute name="HazardousMaterials"
  type="Id"
  isRefinable="yes">
  <Name xml:lang="en">Hazardous Materials</Name>
  <Name xml:lang="fr">Matériaux Dangereux </Name>
</TypeAttribute>
<!-- Below is another endorsed supplemental attribute -->
<TypeAttribute name="ContractId"
  type="Id"
  isRefinable="yes">
  <Name xml:lang="en">Contract ID</Name>
  <Name xml:lang="fr">Contract ID</Name>
</TypeAttribute>
</Type>
</TypeDefinition>

```

CIF Catalog

```

CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item
Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer
Name, Supplier URL, Manufacturer URL, Market Price, Catalog_Item_Source,
HazardousMaterials, ContractId
ITEMCOUNT: 1
DATA
599238, 66056, 66056, "Can of hydrogen", 99998050, 2.95, EA, 2, GasesRus, , ,
4.95, 123ABC, Flammable, Contract-789
ENDOFDATA

```


cXML Catalog

```

?xml version="1.0" encoding="UTF-8"?
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index>
  <IndexItem>
    <IndexItemAdd>
      <ItemID><SupplierPartID>599238</SupplierPartID></ItemID>
      <ItemDetail>
        <UnitPrice><Money currency="USD">2.95</Money></UnitPrice>
        <Description xml:lang="en">Can of hydrogen</Description>
        <UnitOfMeasure>EA</UnitOfMeasure>
        <Classification domain="UNSPSC">99998050</Classification>
        <ManufacturerPartID>66056</ManufacturerPartID>
        <ManufacturerName>GasesRus</ManufacturerName>
        <Extrinsic name="CatalogItemSource">123ABC</Extrinsic>
        <Extrinsic name="HazardousMaterials">Flammable</Extrinsic>
        <Extrinsic name="ContractID">Contract-789</Extrinsic>
      </ItemDetail>
    </IndexItemAdd>
  </IndexItem>
</Index>

```

Chapter 5

CIF Catalog Format

This chapter describes CIF (Catalog File Format) for anyone creating, maintaining, or importing catalog files. It covers both CIF 2.1 and CIF 3.0.

This chapter discusses the following topics:

- [Overview](#)
- [Basic Format](#)
- [CIF Index Files](#)
- [CIF Contract Files](#)
- [Examples](#)
- [Editing CIF Files](#)

Overview

This section introduces CIF, and describes CIF versions, cXML syntax, and file name restrictions.

CIF Versions

CIF 3.0 is an extension of CIF 2.1. The formats are backward and forward compatible: a given CIF parser should be able to read both formats.

CIF 2.1 is an easy-to-use format with a fixed set of 12 fields.

CIF 3.0 offers more fields, which allows you to provide more details or peripheral information about your products and services. For example, each line item can have descriptions in multiple languages and prices in multiple currencies. Also, line items can have parametric data, which allows buyers to perform intelligent searching based on item properties.

For a more general introduction to the fields in a CIF file, see Chapter 2, “Catalog Features”

File Names

You can use any file name for catalogs. However, file names must contain only the following ASCII characters:

- A – Z
- a – z
- 0 – 9
- period (.), dash (-), underscore (_)

Use the extension `.cif` for CIF catalogs.

Use the same file name for all versions of a given catalog. Using the same name makes catalog management easier for buying organizations.

Basic Format

Every CIF file has two sections: *header* and *body*. The *header* specifies general attributes that apply to all the data in the file. The *body* describes the items in the catalog.

CIF Header Syntax

The CIF header defines attributes that apply to the entire file. The following example shows a CIF 3.0 header:

```
CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
COMMENTS: Sample CIF 3.0 catalog
CHARSET: 8859_1
ITEMCOUNT: 20
TIMESTAMP: 2000-04-18 00:00:00
```

FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer Name, Supplier URL, Manufacturer URL, Market Price, Currency, Expiration Date, Effective Date, Classification Codes, Parametric Data, Parametric Name

The first line declares the CIF version, and it must be at the beginning of every CIF file. The remaining lines are optional and can appear in any order. Note that CIF 2.1 files have the time stamp and ITEMCOUNT in the trailer, not the header.

Every line in the header (except the first line) has the format

KEY: value

Keys are case-insensitive, but values are case-sensitive.

- For a list of header keys in CIF index files, see “[Header of an Index File](#)” on page 89.
- For a list of the header keys in CIF contract files, see “[Header of a Contract File](#)” on page 95.

CIF Body Syntax

The body of a CIF file is a series of lines, each in Comma Separated Value (CSV) format.

The body is delimited by the keywords DATA and ENDOFDATA.

Basic Syntax

The basic syntax rules are:

- 1 Each new line is a new entry.** All white space is significant. Do not include spaces, unless they are within quotes (“”). Blank lines are not allowed.

Line returns are not allowed within quoted areas. They are allowed only at the end of each line, which starts a new line. Use a line return at the end of each body line.

- 2 Commas are significant.** Each comma (,) means “go to the next field.” If you have a field that contains a comma, you must quote that field. For example, if you wanted to include the following text in a field:

hammer, wood

surround the whole field in quotation marks:

Field1, "hammer, wood", Field3

- 3 Escape quotes in quoted fields.** To use a quote (") within a quoted field, use two quotes. If the field is not quoted, do not escape the quote. Both of the following examples are legal:

Field1, This flashlight is 5" long, Field3, Field4
Field1, "Flashlight, 5"" long", Field3, Field4

Single quotes (') need no special treatment.

- 4 You can leave fields out.** To leave a field out, put two commas immediately next to each other, with no space in between, as shown below:

Field1, Field2, , Field4

Body Fields

Each row must have the same number of fields. To leave a field empty, enter nothing in the field, just the separating commas. For example:

Field1, , Field3,

CIF 2.1 has 12 fields, in a fixed order.

CIF 3.0 has those same 12 fixed fields, but offers additional optional fields, which can appear in any order after the fixed set of fields. You name the fields in a FIELDNAMES field in the header. The names of these additional fields are fixed; see the FIELDNAMES description on [page 90](#).

You can add your own custom fields (supplemental attributes) to CIF 3.0 catalogs:

- Ariba SN accepts CIF 3.0 catalogs that have any number of fields.
- Most procurement applications recognize only the fields listed in this document. They discard unrecognized fields during catalog import.
- Ariba Buyer 7.0.6 and later can be configured to recognize custom fields as supplemental attributes. For more information, see [“Supplemental Attributes”](#) on page 44.
- Catalog lines must be less than eight kbytes long.

Trailer

The last line in the body is ENDOFDATA. Applications look for this keyword to determine whether the file has been truncated.

CIF 2.1 files can optionally have timestamp and ITEM COUNT lines after the ENDOFDATA line:

```
ENDOFDATA
5/15/2000 15: 25: 04
ITEMCOUNT: 34
```

For CIF 3.0 files, applications ignore any data after the ENDOFDATA line.

Decimal Values

Numbers can use either a period (.) or a comma (,) as the decimal separator; however, to use a comma, quote the entire number. For example, if you want to specify a price of three dollars, you could write either 3.00 or "3,00".

Do not use thousands separators, such as comma, period, or space. For example, to write three thousand dollars, write 3000.00 or "3000,00" but not "3,000.00".

Do not include currency symbols, such as \$, ¥, or £.

Procurement applications determine how many digits appear to the right of the decimal point; for example, 8.50 or 8.500.

Hash Tables

CIF 3.0 uses *hash tables* to list multiple product classification codes, parametric search data, and price segments.

A hash table is represented as a string that specifies a list of key-value pairs in the following format:

```
{ key=value; [key=value; ] * }
```

where both key and value are strings. For example:

```
{TYPE=SEAMLESS; SIZE=20; WEIGHT=94.00; }
```

Hash tables have the following quoting requirements:

- If spaces appear within a key, you do not need to quote the key:

```
{HAZARDOUS MATERIAL CODE=3; }
```

- If spaces appear within a term, quote the term:

```
{CONNECTOR="BUTTRESS END"; }
```
- If commas (,) appear anywhere within the field, quote the entire field:

```
"{TYPE=ROUND, FLAT; }"
```
- If a quote (") appears within a quoted field, escape it with an additional quote:

```
"{TYPE=""ROUND, FLAT""; }"
```
- If the characters ; , } " or \ appear within terms, escape them. You can escape the entire term by enclosing it in quotes:

```
{TYPE="ROUND; {FLAT}"; }
```

Or, you can escape single characters (except commas) within terms with a single backslash (\):

```
{TYPE=ROUND\; FLAT; }  
{TYPE=ROUND 5\"; }
```
- If backslashes (\) or quotes (") appear in terms, escape them with a backslash, regardless of whether they are in a quoted region:

```
{TYPE="ROUND\\FLAT"}; }  
{TYPE="ROUND 5\\""; }
```

CIF Index Files

CIF index files list items, their descriptions, and prices.

CIF 3.0 supports optional separate contract files containing pricing information; for more information, see “[CIF Contract Files](#)” on page 95.

- [Header of an Index File](#)
- [Body of an Index File](#)

Header of an Index File

The following table describes the keys recognized in the CIF index file header. If other keys are present, they do not cause errors, but they are ignored.

Required	Key name	Type	Description	More Info	Example
•	CIF_I_V3.0		Specifies file format and version. For CIF 2.1, use CIF_I_V2.1.		
	CURRENCY	String	Specifies a default currency for interpreting prices in this file. Must be an ISO 4217 currency name. Individual items can override this value by using the Currency body field.	page 40	USD
	CHARSET	String	Specifies the ISO character set used in this file. For valid values, see “Character Set” on page 64. This attribute is recognized only in CIF 3.0 format.	page 64	8859_1
	CODEFORMAT	String 64	Specifies the default commodity code system; usually UNSPSC. If no value is specified, the default coding system is assumed to be SPSC. Ariba Buyer 8 recognizes an optional version number. You can supplement individual items with additional codes by using the Classification Codes field.	page 20	UNSPSC UNSPSC_V7.1 UNDPUNSPSC
	COMMENTS	String	Arbitrary text. This attribute is recognized only in CIF 3.0 format. Comments can contain multiple lines.		
	DUNS	Bool	TRUE or FALSE Indicates whether supplier IDs are D-U-N-S® numbers. This attribute is available for backward compatibility. New catalogs should use the SUPPLIER_ID_DOMAIN attribute instead. Ariba Buyer 8 does not recognize this attribute.		TRUE

Required	Key name	Type	Description	More Info	Example
	FI ELDNAMES	String	<p>Declares the columns in the body of the file.</p> <p>If you do not include FI ELDNAMES, the body must contain only the first 12 (CIF 2.1) fields.</p> <p>If you do include FI ELDNAMES, you must declare all of the first 12 (CIF 2.1) fields; fields beyond the first 12 are optional and can be declared in any order. The names must match the names of the available fields exactly, including capitalization.</p> <p>You can declare your own custom fields, but procurement applications ignore them.</p> <p>This attribute is supported only in CIF 3.0.</p>	page 86	Supplier ID, Supplier Part ID, Manufacturer Part ID, Item Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer Name, Supplier URL, Manufacturer URL, Market Price, Tier, Short Name, Language, Parametric Data, Parametric Name
	ITEMCOUNT	Integer	<p>Number of items in this file.</p> <p>This attribute is supported in the trailer for CIF 2.1, and is optional.</p>		321
	LOADMODE	Char	<p>Declares whether the catalog is complete or an update. Specify F (Full) or I (Incremental).</p> <p>CIF 2.1 is always full; CIF 3.0 is incremental by default.</p> <p>Ariba Buyer 8 and AM-NE perform only full, not incremental.</p>	page 54	F
	SUPPLIERID_ DOMAIN	String	<p>Specifies the domain for supplier IDs used in this file. Typical domains are DUNS and NetworkId.</p>	page 18	DUNS
	TI MESTAMP	String	<p>YYYY-MM-DD hh:mm:ss, where the time is in 24-hour format.</p> <p>For CIF 2.1, this attribute is supported in the footer.</p>		2000-04-18 14: 30: 00
	UNUOM	Bool	<p>Indicates whether this file uses United Nations Units of Measure (UNUOM).</p> <p>If set to a value other than TRUE or if left off, the file is assumed to use American National Standards Institute (ANSI) units of measure.</p>	page 26	TRUE

Body of an Index File

The following table describes the fields that can appear in the body of a CIF file.

The first 12 fields are required and must appear in order. Some of these fields can be empty, but they must be delimited by commas. All fields after 12 (marked with * as the field number) are optional ones available in CIF 3.0.

	Required Value	Field Name	Type and Length	Description	More Info	Example
1	•	Supplier ID	String CIF 2.1: 15 CIF 3.0: 255	Typically a D-U-N-S or NetworkId number. Ariba Buyer supports up to 128 characters. Ariba Buyer 8 accepts only one supplier per catalog. This string is case-insensitive.	page 18	942888710
2	•	Supplier Part ID	String CIF 2.1: 25 CIF 3.0: 255	Part identifier used by supplier. Ariba Buyer supports up to 225 characters. This string is case-insensitive.		SUX-600
3		Manufacturer Part ID	String CIF 2.1: 25 CIF 3.0: 255	Part identifier used by manufacturer. Ariba Buyer supports up to 225 characters. This string is case-insensitive.		286-33
4	•	Item Description	String CIF 2.1: 240 CIF 3.0: 2000	Description of the product or service. Ariba Buyer R7 supports up to 2000 <i>bytes</i> .	page 19	"Super Duper 33-MHz 286, 4MB RAM, 10MB HD, 14" VGA Color Monitor"
5	•	SPSC Code	String 40	Product classification code; usually a UNSPSC code. The CODEFORMAT key in the header dictates the domain. The field is named "SPSC Code" for backward-compatibility; it works with any domain, such as UNSPSC. Each item must have at least one code, either here or in the Classification Codes field. Ariba Buyer 6 and later accept up to 128 characters. This string is case-insensitive.	page 20	43171803

	Required Value	Field Name	Type and Length	Description	More Info	Example
6	•	Unit Price	Decimal	Customer's price. Use either a period or a comma as a decimal separator. Quote the field if using a comma. Do not include currency symbols. Do not use thousands separators, such as commas, periods, or spaces. Procurement applications support at least four digits to the right of the decimal point. No value required for punch-out items.	page 31	1259.59 "17,00"
7	•	Unit of Measure	String CIF 2.1: 40 CIF 3.0: 32	UN or ANSI X.12 standard unit of measure. For PunchOut items, no value is required; this field is ignored. This string is case-insensitive.	page 26	EA
8		Lead Time	Integer	Number of business days between order receiving and delivery to customer.	page 42	3
9		Manufacturer Name	String CIF 2.1: 80 CIF 3.0: 255	Name of the product's manufacturer.		"Super Duper Computer"
10		Supplier URL	String CIF 2.1: 240 CIF 3.0: 255	Web address of the supplier's data sheet.	page 43	http://www.slowcomp.com/products/indepth.htm
11		Manufacturer URL	String CIF 2.1: 240 CIF 3.0: 255	Web address of the manufacturer's data sheet.	page 43	
12		Market Price	Decimal	List price or suggested retail price. This field has the same requirements as Unit Price.	page 31	1599.49
*		Tier	String 128	Name of the tier category for the item. Use the tier field to make items visible only to member organizations in the named tier. Tier values are case-sensitive. For use only with Ariba Marketplace, Network Edition.	page 35	Platinum

	Required Value	Field Name	Type and Length	Description	More Info	Example
*		Short Name	String 50	Short title for the item, to be used in lists where Item Description is too long.	page 19	Laptop Computer
*		Language	String 255	Java locale code for the language used in the Item Description. Use an underscore (_) instead of a dash (-). If you have multiple items in different languages, differentiate them by using the locale code as the Supplier Part Auxiliary ID. This string is case-insensitive.	page 59	fr_FR
*		Currency	String 32	ISO currency code. This value overrides Currency in the header. This string is case-insensitive.	page 40	USD
*		Expiration Date	Date	Date that the item is no longer valid, in YYYY-MM-DD format. This date must be in the future.	page 42	2003-12-01
*		Effective Date	Date	Date that the item becomes valid, in YYYY-MM-DD format. This date must be in the future.	page 42	2003-01-25
*		Classification Codes	String 64	Hash table of domain/value pairs. The procurement application uses the first key that it recognizes. Ariba Buyer 8 recognizes an optional version number. This string is case-insensitive.	page 25 page 87	{ACME=8BE100}
*		Parametric Data	String 255	Hash table of key/value pairs. Keys (parametric attribute names) can be up to 128 characters. The values cannot be null. Parametric data is case-insensitive.	page 48 page 87	{TYPE=SEAMLESS ; SIZE=20: "WEIGHT PER FOOT"=94.00; }

	Required Value	Field Name	Type and Length	Description	More Info	Example
*		Parametric Name	String Buyer 7.0: 24 Buyer 7.1 and later: 128	Textual label for the type of data you supplied in Parametric Data field. Required if you specify Parametric Data. This string is case-insensitive.	page 49	PI PES
*		PunchOut Enabled	Boolean	Specifies a PunchOut catalog item. True, true, False, false, t, f, 0, 1 If true, Supplier URL becomes the URL of the PunchOut catalog; Unit Price, Unit of Measure, and Lead Time are ignored. The default value is False. This value is case-insensitive.	page 27	true
*		Territory Available	String	List of one or more ISO country or region codes. If more than one code, separate them with commas and quote the entire field. This string is case-insensitive.	page 38	"US, GB, IT"
*		Supplier Part Auxiliary ID	String 255	Optional auxiliary string. Part of the unique ID for an item, when combined with Supplier ID and Supplier Part ID. This string is case-insensitive.	page 33	red
*		Delete	Boolean	Optional key to indicate whether this item is to be deleted. True, true, False, false, t, f, 0, 1 The default value is False. This value is case-insensitive. Ariba Buyer 8 loads only full catalogs, so item deletion is not supported.	page 54	t

*. Optional fields that can be used only if they appear in FIELDNAMES in the header.

All fields are case insensitive, except the Tier field in AM-NE.

Catalogs can contain additional supplemental attribute fields. For more information, see [“Supplemental Attributes”](#) on page 44.

CIF Contract Files

CIF contract files contain only item pricing information. Each item in the contract file must match an existing catalog item defined in a CIF index file, using Supplier ID, Supplier Part ID, and optionally Supplier Part Auxiliary ID.

CIF contract files are supported only in CIF 3.0.

Note: Ariba SN, AM-NE, and Ariba Buyer 8 do not support contract files.

The following example shows a simple CIF contract file:

```
CIF_I_V3.0
SEMANTIC: CONTRACT
FIELDNAMES: Supplier ID, Supplier Part ID, Unit Price, Price Segment
DATA
AN0012345, ACF-100, 2.85, {PI ant12=2.75; PI ant14=2.70;}
AN0012345, ACF-200, 3.45, {PI ant12=3.35; PI ant14=3.40;}
ENDOFDATA
```

Header of a Contract File

The following keys are recognized in the CIF contract file header:

Required	Key name	Type	Description	More Info	Example
•	CIF_I_V3.0		Specifies file format and version. CIF 2.1 does not support contract files.		
	CHARSET	String	Specifies the ISO character set used in this file. For valid values, see “ Character Set ” on page 64. This attribute is recognized only in CIF 3.0 format.	page 64	8859_1
	CURRENCY	String	Unique name of a currency defined in the procurement application. The default is USD.	page 40	USD
	DUNS	Boolean	TRUE or FALSE Indicates whether supplier IDs are D-U-N-S numbers. This attribute is for backward compatibility. New catalogs should use SUPPLIER_ID_DOMAIN instead.	page 18	TRUE

Required	Key name	Type	Description	More Info	Example
•	SEMANTIC	String	Always set to Contract, to identify this as a contract file.		SEMANTIC: CONTRACT
	SUPPLIERID_DOMAIN	String	Specifies the domain for supplier IDs used in this file. Typical domains are DUNS and NetworkId.	page 18	DUNS
	COMMENTS	String	Arbitrary text.		
•	FIELDNAMES	String	Comma-separated list of field names. The field names must match exactly, including capitalization.		Supplier ID, Supplier Part ID, Unit Price, Price Segment

Body of a Contract File

Most of the fields in the body of contract files are the same as those in the body of index files. Price Segment appears only in contract files.

The following fields can appear in each line of a CIF contract file:

	Required	Field Name	Type and Length	Description	More Info	Example
1	•	Supplier ID	String 255		page 18	AN0012345
2	•	Supplier Part ID	String 255			SUX-33
3		Supplier Part Auxiliary ID	String 255		page 33	1234
4	•	Unit Price	Decimal		page 31	1259.59
5		Price Segment	String 255	Hash table of key/value pairs.	page 32 page 87	{Plant12 = 2.85; }

Examples

- [Minimal Number of Fields](#)
- [Using 12 Basic Fields](#)
- [PunchOut Index Example](#)
- [Multinational Example](#)
- [Parametric Data Example](#)

Minimal Number of Fields

The following example shows a four-item CIF file, using the minimum number of fields:

```
CIF_I_V3.0
DATA
942888711,100,,"Blue Ballpoint Pen",1213376,1.95,EA,,,,,
942888711,101,,"No. 2 Pencil",1213377,1.50,DZN,,,,,
942888711,102,,"Rubber Eraser",1213472,0.25,PK,,,,,
942888711,103,,"Stapler, Standard",1237461,2.95,BX,,,,,
ENDOFDATA
```

Using 12 Basic Fields

The following example shows a three-item CIF file, using the first 12 available fields:

```
CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERIDOMAIN: DUNS
ITEMCOUNT: 3
TIMESTAMP: 2000-05-15 15:25:04
DATA
942888710,34A11,C11,"Eames Chair, Black Leather",11116767,400.00,EA,3,"Fast
MFG",http://www.acme.com/34A11.htm,http://www.mfg.com/C11/inddepth.htm,400.00
942888710,56A12,C12,"Eames Ottoman, Black Leather",11116767,100.00,EA,3,"Fast
MFG",http://www.acme.com/56A12.htm,http://www.mfg.com/C12/inddepth.htm,100.00
942888710,78A13,C13,"Folding Chair, Grey Stackable",11116767,25.95,EA,3,"Fast
MFG",http://www.acme.com/78A13.htm,http://www.mfg.com/C13/inddepth.htm,25.95
ENDOFDATA
```

PunchOut Index Example

The following example shows a CIF file that points to a PunchOut catalog:

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

Multinational Example

The following example shows a CIF file that lists an item in three languages with different currencies. It also specifies the UNSPSC version for Ariba Buyer 8.

```
CIF_I_V3.0
CHARSET: 8859_1
CODEFORMAT: UNSPSC_V7.0
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item
Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer
Name, Supplier URL, Manufacturer URL, Market Price, Language, Currency,
Supplier Part Auxiliary ID
SUPPLIERID_DOMAIN: DUNS
ITEMCOUNT: 3
TIMESTAMP: 2000-05-15 15:25:04
DATA
6565,2B,2B,"Men's black shoes",53111601,54.95,PR,2,,,,,en_US,USD,en_US
6565,2B,2B,"Chaussures noires des
hommes",53111601,119.95,PR,2,,,,,fr_FR,FRF,fr_FR
6565,2B,2B,"Herrenschuhe schwarz",53111601,34.95,PR,2,,,,,de_DE,DEM,de_DE
ENDOFDATA
```

Parametric Data Example

The following example shows a two-item CIF file, with multiple product classification codes and parametric search data:

```

CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
COMMENTS: Sample CIF 3.0 catalog
CHARSET: 8859_1
ITEMCOUNT: 2
TIMESTAMP: 2000-01-18 00:00:00
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item
Description, SPSC Code, Unit Price, Unit of Measure, Lead Time, Manufacturer
Name, Supplier URL, Manufacturer URL, Market Price, Short Name, Currency,
Expiration Date, Effective Date, Classification Codes, Parametric Data,
Parametric Name
DATA
942888710,SUX-33,286-33,"Super Duper 33-MHz 286, 4MB RAM, 10MB, 14"" VGA color
monitor",43171803,1259.59,EA,3,"Super Duper
Computer",http://www.slowcomp.com/products/indepth33.htm,,1599.49,"Desktop
Computer",USD,2000-12-01,2000-03-01,{ACME=8BE100},{TYPE="IBM PC";SPEED="33
MHZ"};COMPUTERS
942888710,SUX-66,286-66,"Super Duper 66-MHz 286, 8MB RAM, 20MB, 17"" VGA color
monitor",5045990402,1699.99,EA,4,"Super Duper
Computer",http://www.slowcomp.com/products/indepth66.htm,,1999.49,"Desktop
Computer",USD,2000-12-01,2000-03-01,{ACME=8BE101},{TYPE="IBM PC";SPEED="66
MHZ"};COMPUTERS
ENDOFDATA

```

The following table lists the first catalog item, broken up for easy reading:

1 Supplier ID	2 Supplier Part ID	3 Manufacturer Part ID	4 Item Description	5 Product Code	6 Unit Price	7 Unit of Measure	8 Lead Time	9 Manufacturer Name	10 Supplier URL
942888710	SUX-33	286-33	"Super Duper 33-MHz 286, 4MB RAM, 10MB, 14"" VGA color monitor"	43171803	1259.59	EA	3	"Super Duper Computer"	http://www.slowcomp.com/products/indepth33.htm

11 Manufacturer URL	12 Market Price	13 Short Name	14 Currency	15 Expiration Date	16 Effective Date	17 Classification Codes	18 Parametric Data	19 Parametric Name
	1599.49	Desktop Computer	USD	2000-12-01	2000-01-01	{ACME=8BE101}	{TYPE="IBM PC";SPEED="33 MHZ"};	COMPUTERS

Editing CIF Files

You can generate CIF files either automatically or manually. Use automatic generation if you have a database management system (DBMS) containing legacy product information. Use manual generation if you have only hardcopy product catalogs, or if you offer only a few products or services.

To manually generate or edit CIF files, you can use any text editor or spreadsheet application, such as Notepad, Wordpad, Excel, or Word. Remember to save your files as plain text, not as native application files. Also, double check that your editor does not remove leading or trailing zeroes, and that it does not modify the headers.

Note: Excel adds extraneous commas in the header and trailer of CIF files, which you must delete with a text editor such as Notepad.

The following instructions describe how to use Excel to edit CIF files:

▼ **To open CIF files in Excel:**

- 1 In Excel, select **Open** and choose the CIF file.
Excel displays the Text Import Wizard.
- 2 Select “Delimited” and click **Next**.
- 3 Select “Comma” as the delimiter and quote (") as the text qualifier. Deselect “Tab”.
- 4 Click **Next**.
- 5 Select all columns (Shift-click), select “Text” as the column data format, and click **Finish**.
Treat all columns as text so that Excel retains any leading or trailing zeroes.
- 6 To make the resulting spreadsheet more readable, select all cells and select **Format > Column > AutoFit Selection**.
- 7 To calculate the number of catalog items for the ITEMCOUNT field, subtract the number of the last line of data from the number of the first line of data, and add one.

▼ **To save CIF files in Excel:**

- 1 Select the **Save As** command.
- 2 Choose “CSV (OS/2 or MS-DOS)” as the file type.
CSV stands for *comma-separated value*.

3 Enter an intuitive name for the file. The recommended name format is <suppliername>_<buyername>, for example workchairs_acme.

4 Click **Save**.

Note: Excel adds extraneous commas in the header and trailer of CIF files, which you must delete with a text editor such as Notepad.

5 Open the file in Notepad or Wordpad, delete the commas from the header and trailer, ensure that leading or trailing zeroes were not deleted, and save the file.

6 Change the file extension from .csv to .cif.

Catalog files must not be saved in raw Unicode format. For more information, see “[Character Encoding](#)” on page 65.

Chapter 6

cXML Catalog Format

This chapter describes the syntax of cXML (commerce eXtensible Markup Language) catalog files.

- [Overview](#)
- [cXML Basics](#)
- [Index Files](#)
- [Contract Files](#)
- [Examples](#)

Overview

This section introduces XML, cXML, and the types of cXML catalogs.

About XML

XML is a general language for defining the syntax of more specific languages. cXML is an XML-based language and follows standard XML syntax.

XML files use tags to label all the kinds of data they contain. These tags make it very easy for diverse applications to use the same file.

For more information about XML, you can go to the following Websites:

www.xml.com
www.xml.org

About cXML

cXML is a language for conveying data related to e-commerce.

About cXML

cXML is an open standard promoted by Ariba. It is designed to communicate the details of e-commerce transactions, including catalogs, supplier information, and purchase orders.

A unique feature of cXML is its support for *PunchOut catalogs*, which are interactive catalogs hosted on suppliers' live Websites.

Because cXML is an XML-based language, you can use a variety of applications to generate it and parse it.

cXML is the ideal format for automatically generated catalogs, and it supports all the catalog features of CIF.

The description of cXML here is appropriate for suppliers who create content for Ariba SN and for catalog administrators working with procurement applications.

This chapter covers only information related to catalogs. However, cXML is also a general-purpose format for conveying information related to e-commerce.

For cXML news and DTDs (Document Type Definitions), see the cXML.org Website:

www.cxml.org

Types of cXML Catalogs

There are two types of cXML catalog files:

- **Index files** list catalog items, their descriptions, and their prices. Most cXML catalogs are index files.
- **Contract files** lists only prices for items. Each item in these files refers to an item in an index file. Distributing pricing information in a separate file allows you to easily manage pre-negotiated customer-specific contract pricing. Ariba SN, Ariba Buyer 8, and AM-NE do not support contract files.

cXML Basics

This section describes the basics of cXML, such as file naming and basic cXML syntax.

File Names

You can use any file name for catalogs. However, file names must contain only the following ASCII characters:

A – Z
 a – z
 0 – 9
 period (.), dash (-), underscore (_)

Use the extension `.xml` for cXML catalogs.

Use the same file name for all versions of a given catalog. Using the same name makes catalog management easier for buying organizations.

File Headers

cXML catalog files have the following basic format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index>
  body
</Index>
```

The header contains the following elements:

- | | |
|----------|--|
| ?xml | Specifies the XML version and character encoding. "UTF-8" (Universal Transformation Format) supports all character encodings currently supported by Ariba applications. You can also use "ISO-8859-1", which corresponds to Western European language characters (Latin 1). For more information, see “Character Set and Character Encoding” on page 64. |
| !DOCTYPE | Specifies the type of cXML file (Index or Contract) and the URL for the cXML DTD. The DTD describes the format and syntax of all types of cXML documents, and it is used by applications to validate files. |

Basic Syntax

This section gives a brief summary of important points about cXML syntax.

- Every cXML element has a start tag, a body, and an end tag, for example:

```
<UnitPrice><Money currency="USD">1.50</Money></UnitPrice>
```

For tags that do not have contents (nothing between `<tag>` and `</tag>`), you can replace the end tag with shorthand (`</>`), for example:

```
<SearchAttribute name="Color" type="string"/>
```

- Tags must be properly nested. That is, each start tag must have a corresponding end tag.
- Attributes are optional name-value pairs. The values must be enclosed in quotes, for example:

```
<properties help="This is some help text."/>
```

- You can add additional data with an explicit `<Extrinsic>` tag. Use this tag for any data that is not part of the cXML specification, for example:

```
<Extrinsic name="ManufacturerURL">http://www.shoemaker.com</Extrinsic>
```

- You can add comments anywhere, within `<!--` and `-->`. For example:

```
<!-- This is a comment and it can be as long or as many lines as you like. -->
```

- Be careful about using embedded quotation marks when they are not part of the syntax (for example, in the Description field.) Items within fields are literal, and do not have enclosing quotation marks. For example, the line:

```
<Classification domain="UNSPSC">"33331111"</Classification>
```

will not match the UNSPSC code 33331111. Instead, do not quote the data:

```
<Classification domain="UNSPSC">33331111</Classification>
```

- User-visible fields have an `xml:lang` language attribute. You can include one Description field per catalog item, for example:

```
<Description xml:lang="fr-FR">Chaussures noires des hommes</Description>
```

Buying organizations can configure their procurement applications to display the appropriate catalog item, based on the locations of users. For more information about multilingual catalogs, see Chapter 3, “Internationalization and Localization”.

To specify languages, use Java locale codes, for example *it-IT*, *fr-FR*, and *en-US*.

For a list of these codes, see:

<http://java.sun.com/products/jdk/1.2/docs/guide/internationalization/doc.html>

Load Mode

All cXML 1.2.007 and earlier catalogs are incremental. cXML 1.2.008 and later defines an optional `loadmode` attribute for catalogs that can be set to "Full" or "Incremental".

Ariba Buyer 7 and earlier supports only incremental catalogs. It interprets all catalogs as incremental and it does not use the `Loadmode` attribute.

Ariba Buyer 8 and AM-NE support only full catalogs. They interpret all catalogs as full and if catalogs use the `Loadmode` attribute, it must be set to "Full".

Index Files

Index files are lists of products or services for sale. Index files include one or more supplier IDs to identify the supplier. However, each catalog can reference only one supplier; multi-supplier catalogs are not supported in cXML.

The body of index files contains one or more `IndexItem` elements. Ariba Buyer 7 and earlier (cXML 1.2.007 and earlier) also supports an optional set of `SearchGroup` elements, which define the parametric search indexes for catalog items.

The following outline describes a typical index file:

header information

```
<Index>
  <SupplierID domain="DUNS">942888710</SupplierID>
  <Comments xml:lang="en-US">Sample cXML/Index</Comments>
  <SearchGroup>
    used only by Ariba Buyer 7 and earlier
    ...
  </SearchGroup>
</IndexItem>
  ...
</IndexItem>
</Index>
```

Every item in an index file is an `<IndexItemAdd>`, an `<IndexItemPunchout>`, or an `<IndexItemDelete>`.

Every `<IndexItem>` tag includes an ID to identify the item being added or deleted. An *add* includes a full description of the item being added; a *delete* contains only the ID of the item to delete.

Adding Items

Every `IndexItemAdd` element contains exactly one `ItemID` element, one `ItemDetail` element, and one `IndexItemDetail` element. The following example shows a complete cXML catalog containing one `IndexItemAdd`:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index>
  <SupplierID domain="DUNS">942888710</SupplierID>
  <Comments xml:lang="en-US">Sample cXML/Index</Comments>
  <IndexItem>
    <IndexItemAdd>
      <ItemID>
        <SupplierPartID>pn12345</SupplierPartID>
      </ItemID>
      <ItemDetail>
        <UnitPrice>
          <Money alternateCurrency="EUR" alternateAmount="2.74"
            currency="ITL">5300.00
          </Money>
        </UnitPrice>
        <Description xml:lang="en">
          <ShortName>Men's Shoes</ShortName>
          Black squeaky spectator shoes with big heels
        </Description>
        <UnitOfMeasure>PR</UnitOfMeasure>
        <Classification domain="UNSPSC">5136030000</Classification>
        <ManufacturerPartID>MBS3.12</ManufacturerPartID>
        <ManufacturerName>Florsheim</ManufacturerName>
        <URL>http://www.quickshoes.com</URL>
        <Extrinsic name="ManufacturerURL">http://www.florsheim.com
        </Extrinsic>
        <Extrinsic name="Tier">Gold</Extrinsic>
      </ItemDetail>
      <IndexItemDetail>
        <LeadTime>10</LeadTime>
        <ExpirationDate>2002-06-01</ExpirationDate>
        <EffectiveDate>1999-01-01</EffectiveDate>
      </IndexItemDetail>
    </IndexItemAdd>
  </IndexItem>
</Index>
```

To declare Supplier URL, use the tag `<URL>`, and to declare Manufacturer URL, use an extrinsic tag. Manufacturer URL is available only as an extrinsic.

The following example shows an `IndexItemPunchout`:

```

<IndexItem>
  <IndexItemPunchout>
    <ItemID> <SupplierPartID>pn12399</SupplierPartID> </ItemID>
    <PunchoutDetail>
      <Description xml:lang="en-US">Ruby Slippers</Description>
      <URL>http://oz.com/Dorothy/shoes/configurator.asp</URL>
      <Classification domain="UNSPSC">5136030000</Classification>
      <ManufacturerName>Wizard Shoes</ManufacturerName>
      <ManufacturerPartID>WSRS1</ManufacturerPartID>
      <ExpirationDate>2010-01-01</ExpirationDate>
      <EffectiveDate>2000-01-24</EffectiveDate>
      <SearchGroupData>
        <Name xml:lang="en">"DRESSSHOES" </Name>
        <SearchDataElement name="SIZE" value=4/>
        <SearchDataElement name="COLOR" value="RUBY RED" />
        <SearchDataElement name="GENDER" value="WOMEN'S" />
      </SearchGroupData>
      <TerritoryAvailability>US</TerritoryAvailability>
      <TerritoryAvailability>GB</TerritoryAvailability>
    </PunchoutDetail>
  </IndexItemPunchout>
</IndexItem>

```

Deleting Items

The following example shows a *delete*, which identifies an item to delete from the procurement application. If you use Supplier Part Auxiliary IDs to add items, include these IDs as well to delete them.

Note: AM-NE and Ariba Buyer 8 accept only full catalog loading, so they do not support item deletion.

```

<IndexItem>
  <IndexItemDelete>
    <ItemID>
      <SupplierPartID>pn12345</SupplierPartID>
      <SupplierPartAuxiliaryID>Box</SupplierPartAuxiliaryID>
    </ItemID>
  </IndexItemDelete>
</IndexItem>

```

Contract Files

Contract files contain pricing information for items defined in index files.

Note: Ariba SN, AM-NE, and Ariba Buyer 8 do not support contract files. The Contract element was deprecated in cXML 1.2.008.

The following example shows a contract file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Contract SYSTEM
"http://xml.cxml.org/schemas/cXML/1.2.007/cXML.dtd">
<Contract effectiveDate="2000-01-01" expirationDate="2004-06-01">
  <SupplierID domain="DUNS">942888710</SupplierID>
  <Comments xml:lang="en-US">
    Sample cXML contract
  </Comments>
  <ItemSegment segmentKey="Detroit">
    <ContractItem>
      <ItemID>
        <SupplierPartID>12345</SupplierPartID>
      </ItemID>
      <UnitPrice><Money currency="USD">7.50</Money></UnitPrice>
      <Extrinsic name="URL">http://www.shoo.com/Products
    </Extrinsic>
    </ContractItem>
  </ItemSegment>
</Contract>
```

<Contract> is a tag that delimits the contract. It can have optional attributes to set an effective date or expiration date.

Three elements can appear in a contract file:

- <SupplierID> identifies the supplier. To use multiple IDs, use multiple <SupplierID> tags, each with a domain attribute, for example:


```
<SupplierID domain="DUNS">1237676456</SupplierID>
<SupplierID domain="NetworkID">AN0012345</SupplierID>
```
- <Comments> contains arbitrary text to include as file documentation.
- <ItemSegment> specifies a pricing segment for this item. For only one pricing segment, use only one <ItemSegment> tag.

Each item segment defines the pricing for one particular named segment. Each <ItemSegment> is a list of contract items that apply within that segment.

Each `<ContractItem>` identifies an item with `<ItemID>` and specifies a price for that item with `<UnitPrice>`. You can optionally set up overrides for any existing fields with `<Extension>` tags.

The following example shows an item segment that specifies price and Manufacturer URL:

```
<ItemSegment segmentKey="Detroit">
  <ContractItem>
    <ItemID>
      <SupplierPartID>12345</SupplierPartID>
    </ItemID>
    <UnitPrice><Money currency="USD">1.50</Money></UnitPrice>
    <Extension name="URL">http://acme.com/</Extension>
  </ContractItem>
</ItemSegment>
```

Examples

- [Add an Item](#)
- [Delete Items](#)
- [PunchOut Index Example](#)

Add an Item

The following example catalog adds one item (with parametric search data). It also specifies a UNSPSC version for Ariba Buyer 8.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index>
  <SupplierID domain="DUNS">942888711</SupplierID>
  <SupplierID domain="psoft">XYZ124</SupplierID>
  <Comments xml:lang="en-US">
    Sample catalog to show how easy it is to use cXML
  </Comments>
  <IndexItem>
    <IndexItemAdd>
      <ItemID>
        <SupplierPartID>pn12345</SupplierPartID>
      </ItemID>
    </IndexItemAdd>
  </IndexItem>
```

```

    <UnitPrice>
      <Money currency="USD">60</Money>
    </UnitPrice>
    <Description xml:lang="en">
      <ShortName>Men's Shoes</ShortName>
      Black squeaky spectator shoes, with a big heel
    </Description>
    <UnitOfMeasure>EA</UnitOfMeasure>
    <Classification domain="UNSPSC_V7.1">5136000</Classification>
    <ManufacturerPartID>MBS3.12</ManufacturerPartID>
    <ManufacturerName>Florsheim</ManufacturerName>
    <URL>http://www.florsheim.com</URL>
    <Extrinsic name="ManufacturerURL">http://www.shoo.com
    </Extrinsic>
    <Extrinsic name="Tier">Silver</Extrinsic>
  </ItemDetail>
</IndexItemDetail>
  <LeadTime>10</LeadTime>
  <ExpirationDate>2004-06-01</ExpirationDate>
  <EffectiveDate>2000-01-25</EffectiveDate>
  <SearchGroupData>
    <Name xml:lang="en">"DRESSSHOE"</Name>
    <SearchDataElement name="SIZE" value="12"/>
    <SearchDataElement name="COLOR" value="BLACK"/>
    <SearchDataElement name="GENDER" value="MEN'S"/>
  </SearchGroupData>
  <TerritoryAvailable>US</TerritoryAvailable>
  <TerritoryAvailable>GB</TerritoryAvailable>
</IndexItemDetail>
</IndexItemAdd>
</IndexItem>
</Index>

```

Delete Items

The following example catalog deletes two items.

Note: AM-NE and Ariba Buyer 8 accepts only full catalog loading, so they do not support item deletion.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.007/cXML.dtd">
<Index>
  <SupplierID domain="DUNS">942888711</SupplierID>
  <SupplierID domain="psoft">XYZ124</SupplierID>
  <IndexItem>
    <IndexItemDelete>
      <ItemID>

```



```

        <SupplierPartID>pn12356</SupplierPartID>
      </ItemD>
    </IndexItemDelete>
  <IndexItemDelete>
    <ItemD>
      <SupplierPartID>pn12357</SupplierPartID>
    </ItemD>
  </IndexItemDelete>
</IndexItem>
</Index>

```

PunchOut Index Example

The following example contains a link to a PunchOut catalog:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index>
  <SupplierID domain="DUNS">942888711</SupplierID>
  <IndexItem>
    <IndexItemPunchout>
      <ItemD>
        <SupplierPartID>pn12399</SupplierPartID>
      </ItemD>
      <PunchoutDetail>
        <Description xml:lang="en-US">Ruby Slippers</Description>
        <Description xml:lang="it-IT">Rossi Pattini</Description>
        <URL>http://oz.com/Dorothy/shoes/red.htm</URL>
        <Classification domain="UNSPSC">5136030000</Classification>
        <ManufacturerName>Wizards Shoes</ManufacturerName>
        <ManufacturerPartID>WSRS1</ManufacturerPartID>
        <ExpirationDate>2010-01-01</ExpirationDate>
      </PunchoutDetail>
    </IndexItemPunchout>
  </IndexItem>
</Index>

```

Index

A

- adding supplemental attributes in CIF 86
- aggregating for multiple suppliers 18
 - in cXML 107
- aggregators 12
- alternate currency and alternate amount 41, 108
- AM-NE search behavior 56
- AN-ID, see NetworkId domain
- ANSI units of measure 26, 90
- Ariba Buyer search behavior 55
- Ariba Supplier Network (Ariba SN) 13
- Ariba Technical Support ix
- ASCII characters for file names 84, 104
- attributes in parametric data 49
- attributes, supplemental 44
- Auxiliary Part ID, see Supplier Part Auxiliary ID

B

- Big5 character set 64
- blank fields in CIF files 86

C

- case sensitivity in CIF 3.0 94
- catalog extrinsics, see supplemental attributes

- catalogs
 - CIF and cXML overview 15
 - distribution through Ariba SN 14
 - Internet catalogs, see PunchOut catalogs
 - names 84, 105
 - pricing features 31
 - PunchOut, see PunchOut catalogs
 - updating 54
 - updating CIF 90
 - updating cXML 109
 - who creates catalogs 12
- CEFACT units of measure 26
- changing catalog formats 16
- character encoding 64, 65
 - in CIF 89
 - in cXML 105
- character set 64
 - CHARSET 84, 89
- CIF 83–101
 - commas 85
 - contract files 95
 - decimal format 87
 - index files 88
 - overview 15
- classification codes 20
 - multiple per item 25
 - UNSPSC 21
- columns, custom, see supplemental attributes
- commas
 - in CIF 85
 - in prices 31
- comments
 - in CIF 89
 - in cXML 106

-
- contract pricing 37
 - CIF contract files 95
 - cXML contract files 110
 - country and region codes 39
 - Cp12xx character sets 64
 - currency codes 40
 - in CIF 89, 93, 95
 - currency symbol 32, 92
 - custom fields, see supplemental attributes
 - cXML 103–113
 - contract files 110
 - index files 107
 - overview 15
- D
- Data Universal Numbering System (D-U-N-S) numbers 18
 - in CIF 89, 95
 - datasheets, linking with URLs 43
 - Date, see Expiration Date or Effective Date
 - days, see lead time
 - deleting catalog items 55
 - in CIF 94
 - in cXML 109
 - delivery options, using Supplier Part Aux ID 35
 - delivery time, see lead time
 - Description 19
 - in CIF 91
 - in cXML 108
 - Document Type Definition (DTD) for cXML 104, 105
 - documentation
 - typographic conventions viii
 - DTD for cXML 104, 105
 - D-U-N-S numbers 18
 - in CIF 89, 95
 - duplicate part numbers 34
- E
- editing CIF files 100
- Effective Date 42
 - in CIF 93
 - in cXML 110
 - 8859 character sets 64
 - e-mailing catalogs 65
 - ENDOFDATA (in CIF file) 86
 - EnumerationValue element in type
 - definitions 75
 - escaping special characters in parametric data 88
 - euro 2002 conversion 41
 - euro currency 41, 108
 - Excel, editing CIF files 100
 - Expiration Date 42
 - in CIF 93
 - in cXML 110
 - extensions (.cif and .xml) for files 84, 105
 - extrinsics, see supplemental attributes
- F
- FIELDNAMES 86, 90
 - fields, adding in CIF 86
 - file headers
 - CIF 84
 - cXML 105
 - file names
 - CIF 84
 - cXML 105
 - format, see CIF or cXML
 - free text searchable fields 55
 - full vs. incremental catalogs 54
 - CIF 90
 - cXML 106
- G
- GB2312 character set 64
 - grouping buyers, see Price Segments
 - grouping parametric data, see Parametric Name
- H
- Hankaku Katakana 65
-

- hash tables in CIF 87
- hazardous material information, linking with URLs 43
- headers, file
 - CIF 84
 - cXML 105

I

- identifying items uniquely 33
- incremental vs. full catalogs 54
 - CIF 90
 - cXML 106
- index files 38
 - CIF 88
 - cXML 107
- indexItemDelete 55
- IndexItemPunchout 108
- internationalization, see localization
- Internet access requirement for Supplier and Manufacturer URLs 44
- ISO
 - country and region codes 39
 - currency codes 40
- ISO-8859-1 character encoding 105
- Item Description 19
 - in CIF 91
 - in cXML 108

J

- Java locale codes 35, 59, 106

K

- Katakana 65
- KSC5601 character set 64

L

- Language 59
 - in CIF 93
 - in cXML 106
- language codes 35, 59, 106

- lead time 42
 - CIF 92
 - in cXML 53, 77, 81, 112
- lettercase in CIF 3.0 94
- line returns in CIF 85
- list price, see Market Price
- load mode 54
 - CIF 90
 - cXML 106
- locale codes 35, 59, 106
- localization ??-41, 59-??
 - character encoding 64, 65, 105
 - currency 40
 - in CIF 93
 - in cXML 106
 - Language 59
 - Supplier Part Auxiliary IDs 35
 - Territory Available 38

M

- Manufacturer and Supplier URLs 43
 - in CIF 92
 - in cXML 108
- manufacturer's suggested retail price (MSRP) 31
- Market Price 31
 - in CIF 92
- marketplace tiers 35
- MS9xx character sets 64
- multiple languages 59
- multiple product classification codes 25
- multiple supplier ID domains 18
- multi-supplier catalogs 18
 - in cXML 107

N

- names
 - file names 84, 105
 - Item Description and Short Name 20
 - naming restrictions in Ariba Buyer 8 68
 - Parametric Name 49

NetworkID domain
 in CIF 91
 in cXML 107
NetworkId domain 18
newlines, see line returns
Notepad, editing CIF files 65, 100
numbers, in CIF files 87

P

Parametric Name 49
 in CIF 94
 in cXML (SearchGroup) 48, 68
parametric searching 45–51
 in CIF 87, 93
 in cXML 48, 68
 recommendations 50
part numbers, duplicate 34
pictures, linking with URLs 43
Price Segment 32
 in CIF 96
 in cXML (ItemSegment) 110
pricing
 auxiliary part IDs 34
 contract 37
 currency 40
 expiration date 42
 market price 31
 price segments 32
 specialized pricing 31–38
 territory available 38
 tiers 35
 unit price 31
product classification codes 20–26
 hash tables in CIF 87
 multiple per item 25
 UNSPSC 21
publishing catalogs 13
PunchOut catalogs
 appearance in procurement applications 27
 described 14
 in CIF 94, 98
PunchOut Enabled 94
PunchOut URL 29
PunchOut, see PunchOut catalogs

Q

quotes
 in CIF files 86
 in hash tables 87

R

range element in type definitions 75
returns, see line returns

S

safety information, linking with URLs 43
searchable fields in catalogs 55
SearchAttribute and SearchGroup 48, 51, 68
searching, enabling parametric searching 45
SelectedItem element 30
SEMANTIC field in CIF contract files 96
Shift_JIS character set 64
shipping options, using Supplier Part Aux ID
 35
Short Name 20
 in CIF 93
 in cXML 108
specialized pricing 31–38
SPSC 89
supplemental attributes 44
supplemental attributes in CIF 86
supplier aggregators 12
Supplier and Manufacturer URLs 43
 in CIF 92
 in cXML 108
Supplier ID 18
 in CIF 91
 in cXML 107
Supplier ID Domain 18
 in CIF 90
 in cXML 110
Supplier Part Auxiliary ID 34
 in CIF 94
 in cXML 109

T

technical support (Ariba) ix

Territory Available 38
 in CIF 94
 in cXML 109
thousands separator (comma, period, or
 space) 92
Tier 35, 92
time
 expiration date 42
 lead time 42
 timestamp in CIF 90
translation, see localization
type definition files 67–81
 attribute naming restrictions 68
 Type element 70
 TypeProvider 71
typographic conventions viii

U

UCS-2 65
Unicode 65
uniquely identifying items 33
Unit Price 31
United Nations
 country codes 39
 currency codes 40
 UNDP 22
 UNSPSC 21
 UNUOM 26
units of measure (UOM) 26
 in CIF 90
 in cXML 108
UNSPSC 21–24
 in CIF 89, 91
 in cXML 108
UNUOM 90
updating catalogs 54
 changing formats 16
 CIF 90
 cXML 109
URL of PunchOut Website 29
URLs, Supplier and Manufacturer 43
UTF-8 65, 105

V

versions of the UNSPSC 23

W

Web access requirement for Supplier and
 Manufacturer URLs 44
Web address
 for PunchOut catalogs 27
 for supplier and manufacturer information
 43
white space in CIF files 85
wildcarding unneeded digits in UNSPSC 24
Word and Wordpad, editing CIF files 100

X

XML 103
xml
 lang attribute 60

Z

Zenkaku Katakana 65

ARIBA, INC.

807 11th Avenue
Sunnyvale, CA 94089 USA
Telephone: 650.390.1000
Fax: 650.390.1100
www.ariba.com

