

SAP Ariba

PUBLIC

Document Version: 2605 – 2026-05

Customer Catalog Format Reference

SAP Ariba solutions

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

SAP

Content

Customer Catalog Format Reference. 5

Customer Catalogs. 6

Customer Catalog Creators. 6

Catalog Customers. 7

Customer Catalog Types and File Formats. 7

Static Customer Catalogs. 7

PunchOut Customer Catalogs. 8

Customer Catalog Formats. 9

SAP Business Network Customer Catalog Features. 10

Glossary of Customer Catalog Terms. 11

Customer Catalog Fields. 14

Examples of Customer Catalogs in SAP Ariba Solutions. 15

Supplier IDs in Customer Catalogs. 16

Supplier ID Domains. 16

Supplier IDs in SAP Ariba Solutions. 17

Multi-Supplier Customer Catalogs. 17

Supplier Part Auxiliary IDs. 18

Customer Catalog Item Descriptions, Short Names, and Keywords. 19

Customer Catalog Item Specifications. 22

Customer Catalog Commodity Codes. 24

UNSPSC. 25

Multiple Commodity Codes. 27

PunchOut Catalog Items. 28

PunchOut URL. 29

SelectedItem Element. 29

PunchOutLevel Attribute. 30

Units of Measure for Customer Catalog Items. 30

Customer Catalog Pricing. 31

Unit Price, Price Amount, and Market Price. 31

Advanced Pricing Details and Fields. 32

Price Flags and Contract Files. 35

Date-Dependent and Tiered Pricing in BMEcat Catalogs. 36

Date and Quantity-Dependent Price Configuration in CIF Catalogs. 37

Lookup Key-Dependent Price Configuration in CIF Catalogs. 39

Order Quantities for Customer Catalog Items. 42

Territory Available Field and Territory Codes for Customer Catalog Items. . . . 43
Currency, Currency Codes, and Alternate Currency for Customer Catalogs. . . . 45
Effective Date, Expiration Date, and Lead Time. . . . 46
Supplier and Manufacturer URLs for Customer Catalog Items. . . . 47
Optional Supplemental Attributes for Customer Catalog Fields. . . . 47
Parametric Search Data. . . . 48
Declaring Parametric Data Fields. . . . 50
cXML Parametric Data Example. . . . 50
CIF Parametric Data Example. . . . 50
Parametric Attribute Names and Data Types. . . . 51
Required for Ordering Attribute. . . . 53
Relevance Ranking for Customer Catalog Items. . . . 53
Partial Items, Required Fields and IsPartial Field, and Enumeration Values. . . . 54
Related Items and Documents in Customer Catalogs. . . . 55
Additional Item Details in BMEcat Catalogs. . . . 59
Multiple Product Classification Systems in BMEcat Catalogs. . . . 60
Non-Standard Customer Catalog Items. . . . 65
Enabling Incremental Loading for Customer Catalog Items. . . . 66
Searchable Fields for Customer Catalog Products and Services. . . . 67
Images for CustomerCatalog Items. . . . 68
Methods for Linking to Images in Customer Catalogs. . . . 69
Thumbnail, Detail, and Full-Size Images in CIF and cXML Catalogs. . . . 71
Thumbnail, Detail, and Full-Size Images in BMEcat Catalogs. . . . 71
Multiple Images in CIF Catalogs. . . . 72
Format, Size, and Composition for Customer Catalog Images. . . . 74
Website Security for Customer Catalog Images. . . . 75
Thumbnail, Detail, and Full-Size Images in Excel Catalogs. . . . 75
Item Masters in Customer Catalog. . . . 75
Hazardous Material Labels 76

CIF Catalogs. . . . 81
Basics of CIF Catalog Files. . . . 81
CIF Header Syntax and Keys. . . . 82
CIF Body Syntax. . . . 85
Required and Optional CIF Body Row Fields. . . . 87
Special Field Syntax Rules for CIF Catalogs. . . . 95
Supplier IDs in Customer Catalogs. . . . 96
Supplier ID Domains. . . . 96
Supplier IDs in SAP Ariba Solutions. . . . 97
Multi-Supplier Customer Catalogs. . . . 97
Supplier Part Auxiliary IDs. . . . 98
CIF Catalog Examples. . . . 99

Minimum Number of Fields. . . . 99
Example of Using all 12 Basic Fields. . . . 99
Example of PunchOut Index Catalog. . . . 100
Multinational Example. . . . 100
Parametric Search Data Example. . . . 100
Relevance Ranking Example. . . . 101
CIF File Editing. . . . 102
Saving an Excel file to CSV Format for Import to CIF. . . . 103
Opening a CIF file in Excel for Editing. . . . 103

cXML Catalogs. . . . 105
cXML Basics. . . . 105
cXML Catalog Files. . . . 106
cXML Headers in Customer Catalog Files. . . . 107
cXML Catalog Body. . . . 108
cXML Catalog Examples. . . . 111

BMEcat Catalogs. . . . 113
BMEcat and Catalog Example. . . . 113
Supported BMEcat Elements and Attributes. . . . 122
Body elements and attributes. . . . 125

Customer Catalogs in Microsoft Excel (.XLSX) Format. . . . 139

Customer Catalog Internationalization and Localization. . . . 147
Customer Catalog Language. . . . 147
Creating Multilingual CIF and cXML Catalogs. . . . 148
Simple Multilingual Customer Catalogs. . . . 148
Multilingual Customer Catalogs with Territory Available Field. . . . 150
Multilingual Customer Catalogs Combining Languages on One Line. . . . 151
Customer Catalog Character Set and Character Encoding. . . . 153

Type Definition Files. . . . 156
Overview of Type Definition Files. . . . 156
Type Definition File Format. . . . 158
Primitive Types. . . . 161
Enumerated Values in TypeAttribute. . . . 163
Canonical Values in TypeAttribute. . . . 168
Range Values in TypeAttribute. . . . 168
Parametric Data Examples. . . . 169
Parametric Data Example Defining Two Attributes. . . . 169
Parametric Data Example Defining Three Attributes. . . . 170
Supplemental Attribute Examples. . . . 172

Customer Catalog Format Reference

This guide introduces CIF, cXML, BMEcat and Microsoft Excel customer catalog formats in SAP Ariba, with detailed descriptions of customer catalog fields.

Customer catalogs are text files that describe the products and services offered by suppliers.

This guide applies to:

- SAP Business Network
- SAP Ariba Catalog
- SAP Ariba Buying
- SAP Ariba Buying and Invoicing
- SAP Ariba Invoice Management
- SAP Ariba Contract Invoicing

Related guides

Creating and Managing Customer Catalogs

Customer Catalog Administration Guide for Buyers

SAP Ariba Catalog Content API

Customer Catalogs

Customer catalogs are text files stored on SAP Business Network or on your own company's website that describe the products and services your organization offers and the prices you charge.

Customer catalogs serve as the main channel for communicating this information to your customers. You can create customer catalogs for any product or service, regardless of how it is measured, priced, or delivered. Your customers access your catalogs through their procurement applications to see your product and service offerings.

Good customer catalogs are the key to reducing order errors. Make the contents of your customer catalogs concise, complete, and accurate to allow your customers to easily choose the most applicable products or services.

Customer Catalog Creators

Customer catalog creators are typically either suppliers or buying organizations.

- Suppliers create customer catalogs to make their product and service content available to their customers. Suppliers distribute catalogs through SAP Business Network, email, or FTP.
  Typically, suppliers have a dedicated product information group that maintains product and pricing details and generates customer catalogs as needed. Suppliers can also be aggregators, who collect, manage, and distribute product content from many suppliers. This content is then a multi-supplier customer catalog. However, the customer catalog is still sent as a single-supplier catalog with the aggregator specified as the supplier. For information on multi-supplier customer catalogs, see Multi-Supplier Customer Catalogs [page 17].
- Buying organizations can choose to create customer catalogs as well. Buying organizations might create their own customer catalogs because:
  - They want to consolidate approved products from diverse suppliers and make it available to their users.
  - They have legacy suppliers that do not provide electronic customer catalog files.
  - They want product descriptions to contain proprietary information or to have a specific format.
  - They provide products or services internally.

  Customer catalog administrators of SAP Ariba solutions typically create buyer-produced catalogs.

Related Information

Customer Catalog Types and File Formats [page 7]
SAP Business Network Customer Catalog Features [page 10]

Catalog Customers

Catalog customers are individuals in organizations who use procurement systems, such as SAP Ariba Procurement solutions, to download customer catalogs, view product and service offerings, and place orders.

Procurement systems store your customer catalogs internally. After your procurement system approves your customer catalogs, customers can see your content, choose catalog items, and add them to their purchase requisitions.

Related Information

Customer Catalog Types and File Formats [page 7]
SAP Business Network Customer Catalog Features [page 10]
Glossary of Customer Catalog Terms [page 11]

Customer Catalog Types and File Formats

SAP Ariba Procurement solutions accept the following customer catalog types: Static and PunchOut, and the following customer catalog file formats: CIF (Catalog Interchange Format), cXML (commerce eXtensible Markup Language), BMEcat (Bundesverband Materialwirtschaft, Einkauf und Logistik e. V. catalog), and Microsoft Excel.

Static Customer Catalogs

Static catalogs are the simplest form of a customer catalog. A static customer catalog is a single text file that contains your catalog information. Static customer catalogs can be created as CIF or cXML files.

For each item in a static catalog, simply create a line entry with the desired information. The following example shows a short CIF customer catalog from an office supply company with four items in the customer catalog:

```
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
```

The following table lists the first customer catalog item, displayed 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**
942888711	100	(not used)	"Blue Ballpoint Pen"	1213376	1.95	EA	(not used)

9 **Manufacturer Name**	10 **Supplier URL**	11 **Manufacturer URL**	12 **Market Price**
(not used)	(not used)	(not used)	(not used)

Optionally, you can enter additional information that enables more advanced customer catalog features. For complete information about CIF customer catalog fields, see CIF Catalogs [page 81]

To see examples of cXML customer catalogs, see cXML Catalog Examples [page 111].

To see an example of a BMEcat customer catalog, see BMEcat catalogs [page 113].

When accessed from a procurement application, a static customer catalog displays your product or service details.

PunchOut Customer Catalogs

PunchOut catalogs (cXML) are live, interactive customer catalogs that are hosted on your website instead of being uploaded to SAP Ariba Procurement solutions. PunchOut customer catalogs allow customers to click a link to "PunchOut" or go to your organization's website.

Based on the customer, your website can offer previously agreed-upon contract products, prices, and other specialized information. If you already have an e-commerce website, you can modify it to support PunchOut.

For PunchOut customer catalogs, procurement applications display a button (for example, **Buy from Supplier**) instead of product or pricing details.

When customers click this button, their web browser displays a page from the supplier's website. Depending on how you implement this page, customers can browse product options, specify configurations, and select delivery methods. When finished, they click a button that returns the order information to the procurement application. The products and their prices then appear within the procurement application on the purchase requisition.

BMEcat catalogs do not support PunchOut.

Additional References

- *cXML solutions guide*
- *cXML reference guide* at http://www.cxml.org

Related Information

Example of PunchOut Index Catalog [page 100]

Customer Catalog Formats

SAP Ariba solutions support three industry standard customer catalog formats: CIF, cXML, and BMEcat. All three formats are simple text files that are easy to create. SAP Ariba solution sites configured for managing customer catalogs using the content management system (CMS) also support customercatalog files in Microsoft Excel (.xlsx) format.

Format	Description
CIF	CIF (Catalog Interchange Format) is a comma-separated list of customer catalog items and their attributes. SAP Business Network allows you to publish CIF customer catalog files in an Excel format with a `.xls` extension. SAP Business Network has a built-in Excel(`.xls`)-to-CIF file converter that converts the Excel(`.xls`) files to CIF (`.cif`) files internally. For more information, see *Creating and managing customer catalogs*. For details about CIF catalogs, see CIF Catalogs [page 81].
cXML	cXML (commerce eXtensible Markup Language) format is an XML-based document type definition (DTD) 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 customer catalogs. For details about cXML customer catalogs, see Topics about cXML Catalogs [page 105].
BMEcat	BMEcat (Bundesverband Materialwirtschaft, Einkauf und Logistik e. V. catalog) format is an XML-based DTD designed specifically for catalogs by the German association for materials management, purchasing and logistics. In BMEcat, customer catalog data is described with elements and attributes, using a syntax similar to that of HTML. For details about cXML customer catalogs, see BMEcat Catalogs [page 113].
Excel	SAP Ariba solution sites configured for managing customer catalogs using the content management system (CMS) also support customer catalog files in Microsoft Excel (`.xlsx`) format. SAP Business Network allows you to publish customer catalog files in Microsoft Excel (`.xlsx`) format only to customers who use the content management system (CMS) in SAP Ariba solutions to manage their customer catalogs. For details, see Customer Catalogs in Microsoft Excel (.XLSX) Format [page 139].

Deciding Which Customer Catlog Format to Use

Static and PunchOut customer catalog formats differ in their handling of extended catalog features, the method of integration with your existing content, support for localization, and customer catalog expiration dates.

When deciding which customer catalog format to use, consider the source of your product and service content:

Creation Method	Customer Catalog Format to Use
Manually create customer catalogs from a hardcopy customer catalog or word-processor files	Create a CIF customer catalog. CIF is the most human-readable format and most applications make it easy to export content into CIF files. SAP Ariba recommends using the current version of CIF.
Use spreadsheets or simple text files to store and maintain product and service content	Create a CIF customer catalog. Most applications make it easy to export content in comma-separated value (CSV) format.
Use a Product Information Management (PIM) system	Create a cXML customer catalog. 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
Plan to implement PunchOut customer catalogs in the future	Create a cXML catalog in the interim. Learning cXML early will make the transition to PunchOut customer catalogs easier.
Want to provide customers with flexible, real-time customer catalog content	Implement a PunchOut website, customer catalogs, and a PunchOut index file.

SAP Business Network Customer Catalog Features

SAP Business Network is a web-based service provided by SAP Ariba for connecting suppliers to buying organizations. SAP Business Network enables you to publish customer catalogs to your customers and also routes purchase orders from your customers to you.

You can publish CIF and cXML customer catalogs on SAP Business Network. You can also publish customer catalogs in BMEcat and Excel(.xlsx) formats to customers who use the content management system (CMS) in SAP Ariba solutions to manage their customer catalogs.

SAP Business Network offers the following customer catalog management features.

Feature	Description
Validation	When you upload CIF or cXML customer catalogs to your SAP Business Network account, SAP Business Network automatically performs syntactic validation. SAP Business Network flags errors by line number, so you can quickly find them and fix them.
Versioning	SAP Business Network can store multiple versions of your customer catalogs. When you update customer catalogs by uploading new versions, SAP Business Network stores them all. However, when customer catalog subscriptions are imported into SAP Ariba solutions, only the latest versions of the customer catalog are imported. If you want to use an older version of the customer catalog for the subscription, newer versions of the customer catalog must be deleted until that version is the current version, then republish it into SAP Business Network. For more information on customer catalog versioning, see *Creating and managing customer catalogs*.

Feature	Description
Testing	Your SAP Business Network test account has a customer catalog/order tester. This tester displays your customer catalog contents similar to how a customer would see it. You can examine your item descriptions and other customer catalog fields to see how they appear. You can select customer 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 SAP Business Network. You cannot test customer catalogs published for customers who use the content management system (CMS) available in SAP Ariba Procurement solutions solutions to manage their customer catalog data.
Publishing	You can publish each customer catalog so it is either public (visible to all customers) or private (visible to specific customers). You might want to make separate public and private versions of your customer catalogs: • Create public customer catalogs for manufacturers' suggested list prices so that customers can source new products from you. • Create private customer catalogs for prearranged, contract pricing for specific customers. Customers search for companies, industries, and products to find applicable customer catalogs. They download them and import them into SAP Ariba solutions. Users of SAP Ariba solutions search them for needed products and services.

For complete information about SAP Business Network, see *Creating and managing customer catalogs* in the **Help** > **Documentation** area of your SAP Business Network account.

Related Information

Glossary of Customer Catalog Terms [page 11]

Glossary of Customer Catalog Terms

Term	Definition
BMEcat	Bundesverband Materialwirtschaft, Einkauf und Logistik e. V. customer catalog format is an XML-based document type definition designed specifically for customer catalogs. BMEcat uses elements and attributes, using syntax similar to that of HTML, to specify customer catalog data.
canonical value	A canonical value is a static, unique identifier that is independent of language or locale variations.
CIF	Catalog Interchange Format is a simple comma-separated list of customer catalog items and their attributes.
commodity code	Product commodity codes enable procurement applications to display products and services in a hierarchy that makes sense to users. Each customer catalog item must be tagged with a commodity code that indicates the kind of item it is. The recommended product classification domain is the United Nations Standard Products and Services Code (UNSPSC) system.

Term	Definition
contract file	A contract file is a special catalog file that contains prearranged negotiated prices that are offered to specific customers. Contract files list only prices for items. Each item in these files refers to an item in a customer catalog file. Distributing pricing information in a separate file lets you manage pre-negotiated customer-specific contract pricing.
credential	A credential is used to verify a provider's identity.
cXML	commerce eXtensible Markup Language is an XML-based language designed specifically for communicating e-commerce business documents. cXML uses tags and attributes, using a syntax similar to that of HTML, to specify customer catalog data.
DTD	Document Type Definitions (DTDs) precisely define the syntax of XML languages, such as cXML.
escape	(CIF only) Escape is the way to use a special character as a literal. To do so, you need to put a backslash (\) in front of it. For example, \" lets the double quote be seen as a quote instead of a delimiter for an entry. To escape a quote (") within a quoted field, use two quotes (""). This is referred to as "escaped" because the backslash or quote causes the item after it to be treated as a literal value, rather than the way the character would typically be interpreted.
extrinsics	(cXML only) Extrinsics are custom fields agreed to by a supplier and a buying organization.
full loading	Full loading is the process of loading all items in a customer catalog, whether or not they have changed after the last customer catalog loads into the SAP Ariba solution.
incremental loading	Incremental loading is the process of loading only customer catalog items that have changed (were added, deleted, or altered) after the last customer catalog loads into the SAP Ariba solution.
multilingual catalog	A multilingual customer catalog provides item descriptions in multiple languages. Customers can configure SAP Ariba solutions to display the appropriate language based on users' locales.
multi-supplier catalogs	(CIF only) Multi-supplier customer catalogs contain items from multiple suppliers.
kitting types	A kitting type is a special customer catalog item composed of multiple customer catalog items. Kitting types are defined in a type definition file. These type definition files can be defined by buying organizations, suppliers, or supplier aggregators. An example of a kit is a new employee kit. It might contain a handbook, a coffee mug, and a hat.

hash table	A hash table is a string that specifies a list of key-value pairs. CIF uses hash tables to list multiple product commodity codes, parametric search data, and price segments. The format is:
	 `{ key=value;[key=value;] }`
	 where both key and value are strings. For example:
	 `{TYPE=SEAMLESS;SIZE=20;WEIGHT=94.00;}`

non-standard catalog item	A non-standard customer catalog item lacks attributes that the buyer requires to identify it. An example is a customer catalog item without an explicit part number in the customer catalog file.
parametric data	Parametric data is optional customer catalog data that enables users to perform sophisticated searching based on product-specific properties. Parametric searching helps users find items they need in large customer catalogs of similar products.

Term	Definition
parse	Parsing is the process of dividing computer language statements into parts that can be used by a computer.
partial item	Regular customer catalog items that are able to have certain fields missing such as price. They are typically used for services. Using partial items in catalogs lets customers search for and find items that are not fully specified.
primitive type	Primitive types are the fundamental types that define the data that can be used in a type definition field. They can include types such as Integer or CurrencyCode. More complex types are built from primitive types.
PunchOut customer catalog	A PunchOut catalog is a live, interactive catalog that resides on the supplier's website. PunchOut catalogs communicate with procurement applications over the Internet using cXML, which supports two-way communication of electronic commerce details.
related item	(BMEcat, CIF, and Excel) A related item is another item associated with a main item. Related items can be mandatory items, accessories, spare parts, similar items, or follow-up items.
supplemental attribute	A supplemental attribute is an optional customer catalog field that helps customers make purchasing decisions.
Supplier ID	Supplier ID is a specific kind of Organization ID. Organization ID is the term used in SAP Ariba solution guides.
Supplier ID domain	The `Supplier ID` domain specifies which system the customer catalog is using for the Supplier IDs. Most customer catalogs use **Ariba Network ID** (Network ID) numbers for Supplier IDs or D-U-N-S® (Data Universal Numbering System).
Supplier Part auxiliary ID	A supplier part auxiliary ID is part of the unique ID for an item when combined with `Supplier ID` and `Supplier Part ID`. Collectively, the resulting value must be unique.
type attribute	A type attribute is an element that defines the characteristics of data of a particular type.
type definition file	A type definition file is a file extends the customer catalog format by defining special types that can be used to define parametric search attributes and supplemental customer catalog attributes.
UNSPSC	United Nations Standard Products and Services Code (UNSPSC) system is the recommended product classification domain. As of 2024-05-24, the supported UNSPSC version was updated to v25.9.

Related Information

SAP Business Network Customer Catalog Features [page 10]

Customer Catalog Fields

Examples of Customer Catalogs in SAP Ariba Solutions [page 15]

Supplier IDs in Customer Catalogs [page 16]

Customer Catalog Item Descriptions, Short Names, and Keywords [page 19]

Customer Catalog Item Specifications [page 22]

Customer Catalog Commodity Codes [page 24]

PunchOut Catalog Items [page 28]

Units of Measure for Customer Catalog Items [page 30]

Customer Catalog Pricing [page 31]

Order Quantities for Customer Catalog Items [page 42]

Territory Available Field and Territory Codes for Customer Catalog Items [page 43]

Currency, Currency Codes, and Alternate Currency for Customer Catalogs [page 45]

Effective Date, Expiration Date, and Lead Time [page 46]

Supplier and Manufacturer URLs for Customer Catalog Items [page 47]

Optional Supplemental Attributes for Customer Catalog Fields [page 47]

Parametric Search Data [page 48]

Parametric Attribute Names and Data Types [page 51]

Relevance Ranking for Customer Catalog Items [page 53]

Partial Items, Required Fields and IsPartial Field, and Enumeration Values [page 54]

Related Items and Documents in Customer Catalogs [page 55]

Additional Item Details in BMEcat Catalogs [page 59]

Multiple Product Classification Systems in BMEcat Catalogs [page 60]

Non-Standard Customer Catalog Items [page 65]

Enabling Incremental Loading for Customer Catalog Items [page 66]

Searchable Fields for Customer Catalog Products and Services [page 67]

Images for CustomerCatalog Items [page 68]

Item Masters in Customer Catalog [page 75]

Hazardous Material Labels [page 76]

Examples of Customer Catalogs in SAP Ariba Solutions

Customer catalogs allow you to convey information about your selling model and complex pricing structures. You can also use these fields to differentiate your products and services.

Customer catalog fields describe the items in a CIF, cXML, or BMEcat customer catalog. Many fields are required, and others are optional, allowing you to customize and delineate your items with more precision. Note that not all features are available in all formats, and not all features are available in all SAP Ariba solutions. Incompatibilities are noted.

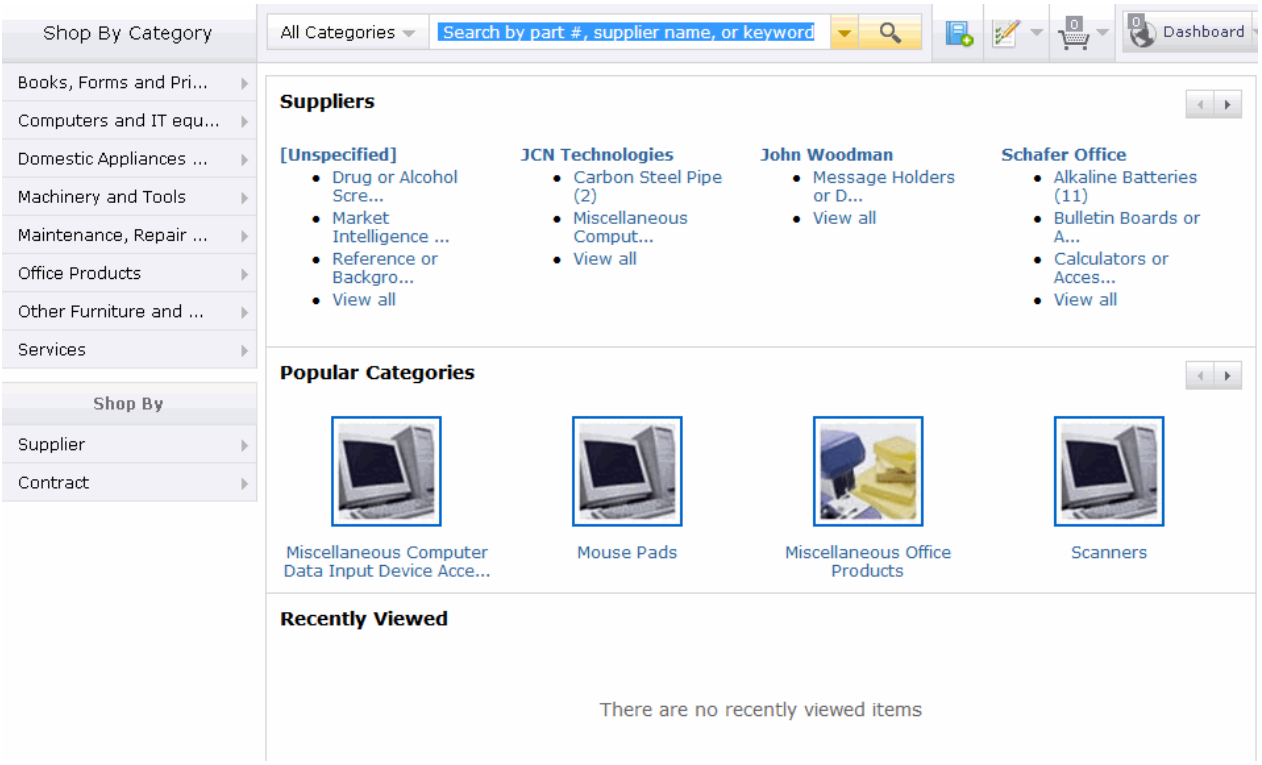
Buying organizations customize the customer catalog item hierarchy that users see. In addition, buying organizations can customize the interface in SAP Ariba solutions, so the actual appearance might be slightly different than these images.

The following examples show how the customer catalog search pages and customer catalog items appear to users of SAP Ariba solutions.

Customer Catalog Search Page

The customer catalog search page can provide a number of different ways for customers to search for products or services, for example, by category, supplier, and recently viewed.

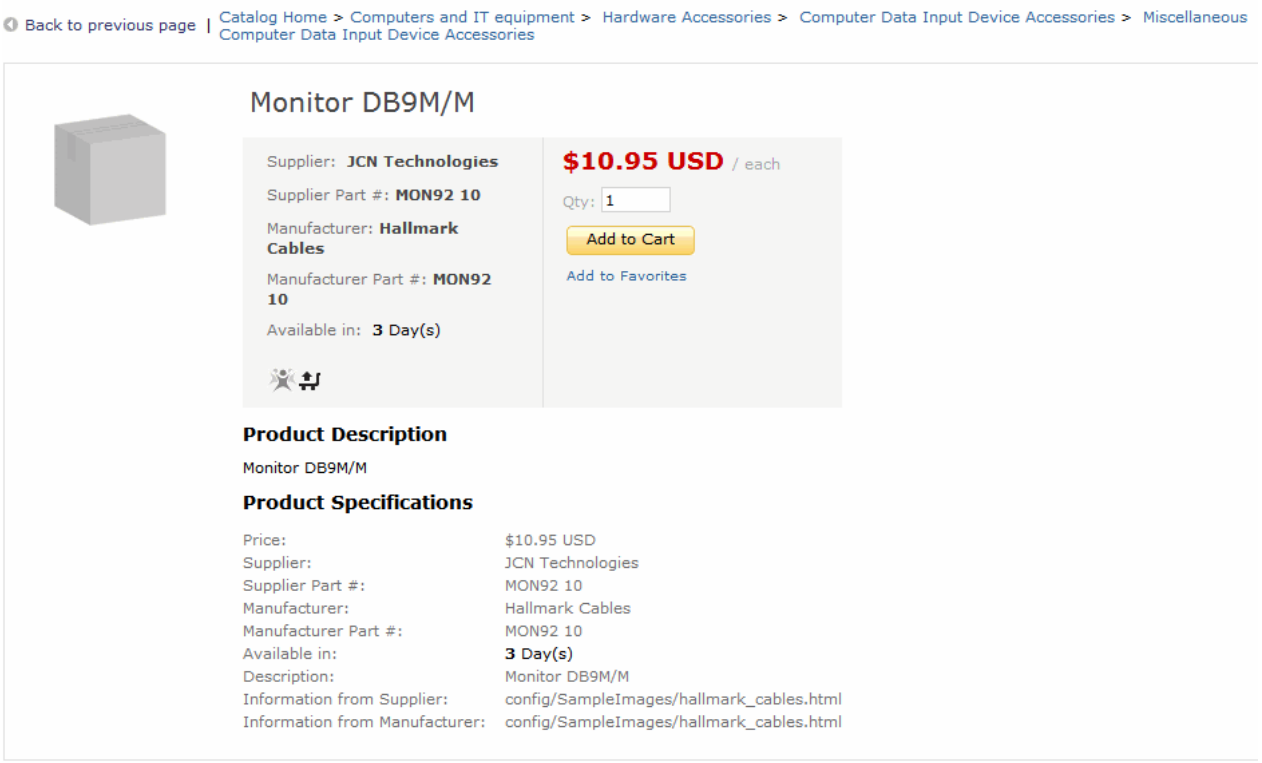
Suppliers can customize the appearance of their customer catalogs, but this figure illustrates a typical customer catalog search page.



Displaying Item Details

The item details page allows you to see specific information about customer catalog items You click an item in the customer catalog search results page to see the item details.

This figure illustrates a typical item details display page.



Supplier IDs in Customer Catalogs

Supplier IDs uniquely identify suppliers. Most customer catalogs use D-U-N-S® (Data Universal Numbering System) or **Ariba Network ID** (NetworkID) numbers for Supplier IDs. You declare which identification system the customer catalog uses, and for each customer catalog item, you identify the supplier that provides that item.

Supplier ID Domains

The Supplier ID Domain is the identification system you use to identify suppliers in your customer catalogs. You can use any identification system, such as D-U-N-S, or customer-specific Supplier IDs.

For easiest integration with procurement applications, use the same Supplier ID Domain that SAP Business Network displays when it lists your account information in CIF and cXML catalogs. For those customer catalogs, domains can also be NetworkIDs.

In BMEcat customer catalogs, the Supplier ID Domain is defined by the supplier ID type, which can be D-U-N-S, ILN, customer-specific, or supplier-specific Supplier IDs.

Supplier IDs in SAP Ariba Solutions

Each SAP Ariba solution maintains a list of known suppliers. During customer catalog import, they match catalog items with a supplier; any items from unknown suppliers produce a warning.

In multi-supplier CIF catalogs, an invalid supplier produces an error and must be fixed and imported again before activating the customer catalog. For more information about multi-supplier CIF catalogs, see Multi-Supplier Catalogs [page 17].

Alternatively, SAP Ariba solutions can query SAP Business Network for the profiles and IDs of new suppliers.

If you distribute your customer catalogs through SAP Business Network, the supplier list within SAP Ariba solutions contains your own Supplier ID. However, if you send customer catalogs through email or FTP, SAP Business Network might not have profiles for your organization, so administrators of SAP Ariba solutions would need to manually enter your information.

> **Note**
>
> Enable the `Application.OrganicGrowth.OrganicGrowthEnabled` parameter for SAP Ariba Buying and Invoicing to automatically generate the following supplier IDs for unknown users who punch in to SAP Ariba Catalog. This occurs when a copy order is received from the SAP Business Network and no matching users who punch in to SAP Ariba Catalog exist in SAP Ariba Buying and Invoicing based on the credentials provided in the purchase order cXML.
>
> - For common users, SAP Ariba Buying and Invoicing creates `Buyer_ System ID`.
> - For partitioned users, SAP Ariba Buying and Invoicing creates `SU_INTERNAL Vendor ID`.
>
> Contact SAP Ariba Support to enable this parameter.

Multi-Supplier Customer Catalogs

A multi-supplier customer catalog is a CIF catalog file that contains items from more than one supplier. Multi-supplier customer catalogs are not supported in cXML or BMEcat.

Manually imported local customer catalogs can contain multiple suppliers if a checkbox in Ariba Administrator is checked during import. SAP Business Network subscriptions (which are automatically loaded into SAP Ariba solutions when a catalog is updated) can contain CIF catalogs with multiple suppliers if the parameter `System.Catalog.Network.DefaultMSCif` is set in SAP Ariba solutions. If you are a supplier and want to publish multi-supplier CIF catalogs on SAP Business Network for customer subscriptions, make sure your customers have this parameter set.

> **Note**
>
> For information about using multi-supplier CIF catalogs with SAP Ariba solutions, see the *Catalog administration guide for buyers*.

The following is an example of a multi-supplier CIF catalog:

```
CIF_I_V3.0
CODEFORMAT: UNSPSC
LOADMODE: F
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
ITEMCOUNT: 3
DATA
29,RAY19844,824-2,"AA Rechargable Batteries 2-PK",26111701,3.02,BOX,0,,,,
599238,66056,66056,"Large Hot/Cold Drinking Cups",99998050,2.95,DZ,2,,,,
98-747-7712,Burgundy-1,Burgundy-1,"1996 Pinot Noir",50202203,25,EA,1,,,,
ENDOFDATA
```

Multi-supplier CIF catalogs are supported by Ariba Buyer 8.2 (and later) and SAP Ariba Procurement solutions. Multi-supplier cXML catalogs are used only in PunchOut scenarios. PunchOut shopping carts (cXML `PunchOutOrderMessage` documents) can refer to multiple suppliers. Catalog aggregators who want to use cXML should consider creating PunchOut sites instead of static customer catalog files. For more information about PunchOut, see PunchOut Catalogs [page 8].

Supplier Part Auxiliary IDs

You must uniquely identify all customer catalog items to prevent conflicts. SAP Ariba solutions identify items by using a combination of their `Supplier ID`, `Supplier Part ID`, and `Supplier Part Auxiliary ID`.

`Supplier Part Auxiliary ID`s are used in the following cases:

- To uniquely identify services, which sometimes do not have part numbers.
- To provide customer catalog items in multiple languages. For example, an item that is available in Italy, France, and Germany can have three `Supplier Part Auxiliary ID`s, which creates three customer catalog items. Each customer catalog item can have a different `Description` field (in Italian, French, or German.) Similarly, each item can have a different `Currency`, `Unit Price`, `Lead Time`, and `Supplier URL`.
  For the `Supplier Part Auxiliary ID`s, use Java locale codes, for example `it_IT`, `fr_FR`, and `de_DE`. For a list of these codes, see:
  http://java.sun.com/j2se/1.4.2/docs/guide/intl/locale.doc.html
  Buying organizations can configure SAP Ariba solutions to determine which customer catalog items to display based on the locations of users. For more information, see Creating Multilingual Catalogs [page 148].
- To differentiate items that have the same `Supplier Part Number`, but different lead times or delivery options, you can use the values `ground`, `2-day,` and `overnight` as `Supplier Part Auxiliary ID`s. (You can also specify a different price for each item.)
- To differentiate items that have the same `Supplier Part Number`, but are packaged differently, 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
  BecauseSAP Ariba solutions do not use `Descriptions` to differentiate items, they interpret these items as duplicates. To prevent duplicate items, assign them `Supplier Part Auxiliary ID`s:
  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

Both items are then considered unique because they have different `Supplier Part Auxiliary ID`s.

> **Note**
>
> Do not use the characters `?`, `{`, and `}` in `Supplier Part Auxiliary ID`s; they cause validation errors.

SAP Ariba solutions enforce a character limit of 255 for the `Supplier Part Auxiliary ID` field.

Customer Catalog Item Descriptions, Short Names, and Keywords

Each CIF or cXML catalog item must have an Item Description and can optionally have a Short Name. Each BMEcat catalog item must have a short name (`DESCRIPTION_SHORT`) and an item description (`DESCRIPTION_LONG`). Procurement applications display these fields to help users decide which products and services to purchase.

SAP Ariba Procurement solutions use short names, descriptions, and keywords as the basis for item search and keyword refinement. Item Description and Short Name 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

We recommend that you do not import customer catalog items with underscore (_) in their names because when users search for such items, the search results might be incorrect.

Item Description

`Item Description` can contain up to 2000 characters for CIF and cXML. `DESCRIPTION_LONG` can contain up to 64000 characters in BMEcat. SAP Ariba solutions allow up to 2000 characters.

If you provide only `Item Description` or `DESCRIPTION_LONG` (without `Short Name` or `DESCRIPTION_SHORT`), procurement applications truncate it to fit product lists and display the truncated value as the Short Name in the user interface. 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 Catalog Language [page 147].

Displaying customer catalog item descriptions as formatted text

Customer catalog item descriptions can be displayed as formatted text in the item details pages of the SAP Ariba solution. Formatting options such as bold and italic font styles, paragraphs, line breaks, and lists are supported in CIF, cXML, and BMEcat catalog files.

The following formatting options are supported:

- paragraphs
- line breaks

- boldface font
- italic font
- ordered list
- unordered list

The item descriptions are displayed as formatted text only in the item details and compare items pages in SAP Ariba solutions, and are displayed as plain text (without the HTML tags) in all other pages where customer catalog item descriptions are displayed. For example, in the customer catalog search results pages, and in purchase requisition and other related documents, the item descriptions are displayed as plain text.

This feature is applicable both to catalogs that are imported to SAP Ariba solutions locally, as well as customer catalogs that are downloaded from SAP Business Network when the *Ariba Network Full Subscription Synchronization* scheduled task is run.

Customer catalog creators can use HTML tags to format the contents of the item description fields in customer catalog files. CIF, cXML, and BMEcat catalog file formats support the following HTML tags:

Formatting option	HTML tags	Description
Paragraph	`<p></p>`	Defines a paragraph.
Line breaks	` ` ` `	Inserts a single line break.
Boldface font	`<b></b>`	Defines bold text.
Italic font	`<i></i>`	Defines italics text.
Ordered list	`<ol></ol>` **Note** Attributes are not supported.	Defines an ordered list.
Unordered list	`<ul></ul>` **Note** Attributes are not supported.	Defines an unordered list.

Note

In cXML catalog files, the item description data must be enclosed within a `CDATA` section. For example, <![CDATA[This is the description of the item]]>.

Consider the following when creating your catalog files:

- All invalid and unsupported tags will be converted to and displayed as plain text in the user interface.
- SAP Ariba solutions ignore supported HTML tags when the customer catalog index is rebuilt, but the unsupported tags are indexed and will be displayed in customer catalog search results as normal text.

- Ensure that you add necessary spaces and punctuation to the item descriptions. No spaces are added when the HTML tags are stripped to display the descriptions as plain text in the customer catalog search results and other pages.
- The HTML tags are retained when customer catalogs are exported from SAP Ariba solutions.
- The HTML tags are also retained when customer catalog data is extracted in JSON format from SAP Ariba solutions.
- The HTML tags are also retained in all punchin request and response documents.

Short Name

`Short Name` or `DESCRIPTION_SHORT` appears in the product lists presented to users. SAP Business Network supports a maximum of 50 characters for this field whereas, SAP Ariba solutions support a maximum of 80 characters.

If you use `Short Name` in your CIF or cXML catalogs, be sure to include the `Short Name` terms in the `Item Description`. If you use `DESCRIPTION_SHORT` in your BMEcat catalogs, be sure to include the `DESCRIPTION_SHORT` terms in the `DESCRIPTION_LONG`. By default, only the `Item Description` or `DESCRIPTION_LONG` is included on purchase orders and sent to suppliers.

For example:

Field	Example
`Short Name/ DESCRIPTION_SHORT`	`SUX-6000 Computer`
`Item Description/ DESCRIPTION_LONG`	`SUX-6000 Computer: Hexapentium-powered 33-Hz desktop computer. Three 10-KB disk drives, four CD-ROM drives, two Zip drives, 5-Mbit/s Ethernet card, and 4 MB RAM. Includes two monitors and a keyboard. Mouse not included.`

Keywords

SAP Ariba Procurement solutions use short names, descriptions, and keywords as the basis for item search and keyword refinement. For CIF and BMEcat catalog items, keywords can be defined explicitly to improve the likelihood of an item being found when users search the catalog.

> **Note**
>
> While the `Keywords` field listed in the catalog file affect the search capability and allows items to be found when searching for those specific keywords, the **Keyword** filter that is in the **Filter By** section might not contain all the keywords listed in the `Keywords` field and instead contain additional keywords that are system-generated.

For more information, see Relevance Ranking for Customer Catalog Items [page 53].

Customer Catalog Item Specifications

In CIF and BMEcat catalogs, the Item Spec customer catalog field is used to specify the additional attributes or specifications of an item that help describe and identify the customer catalog item better. These details are displayed in the **Additional Specifications** section in the **Item Details** pages in the customer catalog user interface.

Item Specifications in CIF Catalogs

The Item Spec customer catalog field is used to specify the additional attributes or specifications of an item. You define additional attribute data as lists of key-value pairs. The syntax is `"{key1=value1;key2=value2;key3=value3;keyn=valuen}"`. Do not enclose keys or values in quotes (" "). For example:

```
{Nominal size= DN 80;Design spec, pressure rating= ASME CL 900;Design spec=
API SPEC 6D\; ISO 14313;Design spec, f to f (e to e)= ISO 14313;Length, f
to f (e to e)= 384 mm;Type of end connection= FLANGED;Type of face= RING
JOINT;Type of pattern= STANDARD;Type of closure member= TRUNNION MOUNTED BALL;Type
of bore= FULL;Type of construction= TWO/THREE PIECE;Type of bonnet/cover= BOLTED
(NON-EXTENDED);Type of seat= SOFT;Type of seat piston= SINGLE;Uni/bi-directional=
BI-DIRECTIONAL;}
```

`Item Spec` field can contain upto a maximum of 6000 characters while each key-value can contain upto 255 characters.

The **Additional Specifications** section in the **Item Details** pages in the catalog user interface displays the key-value pairs as a table. The values in the key-value pairs are searchable in customer catalog search queries.

> **Note**
>
> The Item Spec fields are not displayed in the **Compare Items** pages.
>
> The Item Spec fields are not exported when customer catalogs are exported.

SAP Ariba solutions support the Item Spec field only in sites configured for managing customer catalogs using the content management system (CMS).

Item Specifications in BMEcat Catalogs

The `FEATURE` element within the `ARTICLE_FEATURES` element in BMEcat catalogs describes the additional features or specifications of an item that help describe and identify the customer catalog item better. You can specify multiple features of an item.

The following table lists the elements within the `FEATURE` element:

Element	Mandatory	Type and length	Notes

FNAME	Yes	String (60)	Indicates the unique name of a feature of an item.
			Example:
			`<FEATURE>`
			`<FNAME>Length</FNAME>`
			`<FVALUE>32</FVALUE>`
			`<FUNIT>cm</FUNIT>`
			`</FEATURE>`

FVALUE	Yes	String (60)	Indicates the value of the specified feature. If the value contains any of the following special characters, precede the character with a backslash (\) to avoid catalog validation errors: • / • { • } • ; • = • , • " For example, if the value is `7,4`, specify it as `7\,4`.
FUNIT	No	String (20)	Indicates the unit of measure of the specified feature.

SAP Ariba solutions process the FNAME and FVALUE values that you specify as key-value pairs and map them to the **Item Spec** catalog field by default. The **Additional Specifications** section in the **Item Details** pages in the customer catalog user interface displays the key-value pairs as a table. FUNIT values when specified and automatically appended to the FVALUE value. If you specify multiple FVALUE values, they are displayed in the table seperated by commas.

The following lists a BMEcat catalog that describes the various specifications of an item using the FEATURE element:

```
<ARTICLE_FEATURES>
    <REFERENCE_FEATURE_SYSTEM_NAME>eclass-3.0</REFERENCE_FEATURE_SYSTEM_NAME>
    <REFERENCE_FEATURE_GROUP_ID>24-11-03-21</REFERENCE_FEATURE_GROUP_ID>
    <FEATURE>
        <FNAME>Width</FNAME>
        <FVALUE>15</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
    <FEATURE>
        <FNAME>Height</FNAME>
        <FVALUE>8</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
    <FEATURE>
        <FNAME>Length</FNAME>
        <FVALUE>32</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
    <FEATURE>
```

```
        <FNAME>Colors</FNAME>
        <FVALUE>red</FVALUE>
        <FVALUE>blue</FVALUE>
        <FVALUE>green</FVALUE>
        <FVALUE>yellow</FVALUE>

    </FEATURE>
</ARTICLE_FEATURES>
```

> **Note**
>
> - The `VARIANTS` feature within the `FEATURE`element is not supported.
> - When specifying feature names in the `FNAME` element, names that are defined in `FT_NAME` under the `FEATURE_TEMPLATE` element are not supported.

Customer Catalog Commodity Codes

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

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

- Prevent classes of employees, such as executive assistants, 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.

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

You can use any commodity coding system (called a classification domain) in your customer catalogs. However, both customer catalogs and procurement applications must use the same system, so if you use a non-standard system, you must communicate its details to all customers that use your customer catalogs. You can include multiple commodity coding systems in the same customer catalog so it can be used by diverse customers. If you are part of a buying organization that uses custom commodity codes, you need to communicate those codes to all your suppliers.

In addition to UNSPC, which is described in detail in the following section, other popular commodity coding systems are:

- eCl@ss (Standardized Material and Service Classification)?www.eclass.de/eclasscontent/index.html.en
- NAICS (North American Industry Classification System)?www.census.gov/eos/www/naics/
- ISIC (International Standard Industrial Classification)?unstats.un.org/unsd/cr/registry/isic-4.asp

Related Information

UNSPSC [page 25]
Multiple Commodity Codes [page 27]

UNSPSC

The recommended 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. Customer catalogs that use this system can be used by most customers.

UNSPSC Classification System

The UNSPSC classification system contains descriptions of more than 21,718 products and services. The coding structure is hierarchical, which combines similar products into standardized product groups. 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
```

Examples of UNSPSC Classification Domains

The United Nations Development Programme (UNDP) organization provides UNSPSC code sets.

If there is no UNSPSC code for one of your products or services, you can contact UNDP and request one. Within a short period, the managing organization either creates a new code or recommends an existing number.

The following examples show how to specify UNSPSC 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>
```

BMEcat catalogs use only the UNSPSC classification domain for UNSPSC codes. The classification system in a BMEcat catalog can be specified at either the header or line level. The following examples show both:

BMEcat (header level)
`T_NEW_CATALOG`
	`<CLASSIFICATION_SYSTEM>`
	`  <CLASSIFICATION_SYSTEM_NAME>UNSPSC</CLASSIFICATION_SYSTEM_NAME>`
	`<CLASSIFICATION_SYSTEM>`

BMEcat (line level)
`ARTICLE`
	`<ARTICLE_FEATURES>`
	`  <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</`
	`REFERENCE_FEATURE_SYSTEM_NAME>`
	`  <REFERENCE_FEATURE_GROUP_ID>44111501</`
	`REFERENCE_FEATURE_GROUP_ID>`
	`</ARTICLE_FEATURES>`

UNSPSC Versions

Later UNSPSC versions cover more products and services, reorganize existing commodities, and assign them new codes. SAP Ariba solutions allow catalogs to use UNSPSC versions different from the buying organization.

SAP Ariba solutions use an audit file to translate the domain-specific commodity codes in the catalog to the commodity codes used by their configuration's standard version, called the pinned version. For more information about enabling pinned versions, see the *Customer catalog administration guide for buyers*.

Catalogs created for SAP Ariba solutions must use the UNSPSC version used by the buying organization. Contact your customers to find out the UNSPSC version they use.

SAP Ariba solutions and SAP Business Network provide translation from one UNSPSC version to another, so using the same UNSPSC version as your customers is not required. However, use similar UNSPSC versions whenever possible.

For example:

CIF Header	`CODEFORMAT: UNSPSC`
cXML `ItemDetail`	`<Classification domain="UNSPSC_V7.1">10101516</Classification>`

BMEcat (header level)
`ARTICLE`
	`<CLASSIFICATION_SYSTEM>`
	`  <CLASSIFICATION_SYSTEM_NAME>UNSPSC-7.1</`
	`CLASSIFICATION_SYSTEM_NAME>`
	`<CLASSIFICATION_SYSTEM>`

BMEcat (line level)
`ARTICLE`
	`<ARTICLE_FEATURES>`
	`  <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC-7.1</`
	`REFERENCE_FEATURE_SYSTEM_NAME>`
	`  <REFERENCE_FEATURE_GROUP_ID>44111501</`
	`REFERENCE_FEATURE_GROUP_ID>`
	`</ARTICLE_FEATURES>`

SAP Ariba solutions and SAP Business Network uses this optional version number to identify the UNSPSC version used by the particular buying organization. For more information about these mappings and audit files, see the *Customer catalog administration guide for buyers* or *Creating and managing customer catalogs*.

Commodity Code Wildcarding

To convey a higher-level category or a mix of items, fill unneeded commodity code digits with zeros. This shortening of commodity 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 pair of digits 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.

Multiple Commodity Codes

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

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

Another reason to use multiple commodity codes is to allow the same customer catalog to be used by different customers. For example, a customer catalog of chemicals can use both a broad classification system (such as UNSPSC), and a narrow classification system (such as a chemical industry coding system). The customers that use your customer catalog can use the system that they find the most useful.

To add multiple commodity codes to an item, do one of the following:

- For CIF files, add the multiple commodity codes to the Classification Codes field. The entry for a single commodity code might be `{UNSPSC=55667788;}`, while for multiple commodity codes it might be `{UNSPSC=55667788;ACME=459876;}`.

> **Note**
>
> When adding multiple commodity codes to an item you do not need to define the CODEFORMAT in the CIF file header.

- For cXML files, add a line under the existing commodity code for each additional commodity code. For example,
  ```
  <Classification domain="UNSPSC">55667788</Classification>
  <Classification domain="ACME">459876</Classification>
  ```

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

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.

> **Note**
>
> CIF and cXML support PunchOut catalog items. BMEcat does not.

PunchOut index catalog items have less information in the customer catalog file than static customer catalog items. For example, pricing information in the customer catalog file is ignored; pricing is set by your site for each PunchOut session.

Procurement systems display the following fields for PunchOut index catalog items:

- Supplier Part ID
- Manufacturer Part ID
- Item Description
- Manufacturer Name
- Short Name

Additional References

- *cXML solutions guide*
- *cXML reference guide* at http://www.cxml.org

PunchOut URL

SAP Business Network receives PunchOut requests from customers, validates them, and routes them to your PunchOut site. The method SAP Business Network uses to determine the URL for your site depends on the cXML versions used by your customers and your PunchOut site.

There are two places where you can specify the URL of your PunchOut website. The first method is recommended because if you change any of your URLs, you can easily change your cXML profile without having to sign in to SAP Business Network.

URL Specified in Your cXML Site Profile

If you have a cXML 1.1 or later site, it must support the cXML Profile transaction. Specify a PunchOut URL in the cXML profile, and specify the cXML profile URL in the Configuration area of your SAP Business Network account.

For more information about cXML Profile transactions, see the *cXML Solutions Guide*.

SAP Business Network periodically sends a `ProfileRequest` document query to your cXML site. Your cXML site responds with a `ProfileResponse` document listing URLs for all the cXML requests it supports, including PunchOut requests. SAP Business Network routes PunchOut requests to the `PunchOutSetupRequest` URL listed in your `ProfileResponse`.

URL Specified on SAP Business Network

If you do not specify a cXML site profile URL, you can specify the URL of your PunchOut site in the "PunchOut Setup Request" field in the Configuration area of your SAP Business Network account.

SAP Business Network no longer uses URLs specified within cXML `PunchOutSetupRequest` (from your PunchOut index catalogs). PunchOut sessions are possible only if you specify your PunchOut URL through your cXML profile or your SAP Business Network account.

SelectedItem Element

Some versions of procurement systems use the `SelectedItem` element instead of a URL.

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

- SAP Ariba Procurement solutions 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 customer catalog.
- Some procurement applications do not use `SelectedItem`, so the URL in your customer catalog is the only way to specify the item to punch out for.

For information about the file format of the `SelectedItem` element, see the *cXML Solutions Guide*.

PunchOutLevel Attribute

Suppliers have the option to include the `PunchOutLevel` attribute in CIF or cXML PunchOut index catalogs. The `PunchOutLevel` attribute specifies how to display the item in the procurement system customer catalog.

The `PunchOutLevel` attribute can have the following values, in hierarchical order:

- `store`: Use if the PunchOut item points to a single page that allows users to search for any item. SAP Ariba Buying and Invoicing solution users punch out directly without displaying item details. These items display at the top of categories, above products.
- `aisle`: Use if the PunchOut item groups related items together. SAP Ariba Buying and Invoicing solution users punch out directly without displaying item details.
- `shelf`: Use if the PunchOut item groups similar products together, from which the user must decide when shopping. SAP Ariba Buying and Invoicingsolution displays item details and allows users to punch out.
- `product`: Use if the PunchOut item is for a single product or SKU. SAP Ariba Buying and Invoicing solution displays item details and allow users to punch out.

If the supplier uses no `punchoutLevel` attribute, SAP Ariba solutions consider the item to be a store-level item.

Units of Measure for Customer Catalog Items

Every customer 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 customer catalogs must use a standardized set of units of measure.

The two commonly used standards for units of measures 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:

- United Nations Center for the Facilitation of Procedure and Practices for Administration, Commerce, and Transport (UN/CEFACT) codes
- United Nations Trade Data Elements Directory (UNTDED) common codes
- United Nations Economic Commission for Europe (UN/ECE) Trade Facilitation Recommendation 20–Codes for Units of Measurement used in International Trade

For information on this standard, see:

www.unece.org/cefact/recommendations/rec20.xml

The following table lists frequently used UNUOM codes:

UNUOM Code	Meaning
AY	assembly
EA	each
BX	box
DZN	dozen

UNUOM Code	Meaning
GRO	gross
HUR	hour
KGM	kilogram
LBR	pound
M4	monetary value
RO	roll
PR	pair
PK	pack

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

Customer Catalog Pricing

Customer catalogs contain the prices of items. In the simplest case, there is a price for each item. But pricing is often more complicated, so customer catalog formats have features that support more flexible pricing models. For example, PunchOut items and partial items do not require pricing.

The following specialized pricing features are available:

- Unit Price, Price Amount, and Market Price [page 31]
- Advanced Pricing Details and Fields [page 32]
- Price Flags and Contract Files [page 35]
- Date-Dependent and Tiered Pricing in BMEcat Catalogs [page 36]
- Date and Quantity-Dependent Price Configuration in CIF Catalogs [page 37]

Unit Price, Price Amount, and Market Price

`Unit Price` (CIF and cXML) or `PRICE_AMOUNT` in `PRICE_DETAILS` (BMEcat) is a required field that specifies the cost of an item. `Market Price` (CIF only) is an optional field that specifies the market price, list price, or manufacturer's suggested retail price (MSRP) of an item.

SAP Business Network requires a price in either the `Unit Price` or `Market Price` field. If the unit price is not specified, the available market price is used.

> **Note**
>
> - When using CIF catalogs, if you want to indicate to buyers that an item is an inquiry item, and that they have to contact you for prices, set `Unit Price` to `Zero` and add relevant header-level comments.

- In BMEcat catalogs, SAP Ariba solutions only support the `net_customer` type for `PRICE_DETAILS`; items with other price types are ignored during customer catalog import.

Digit Separators in Prices

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, cXML, and BMEcat:

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

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

SAP Ariba solutions 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 in Prices

Do not include currency symbols such as $, ¥, or £. in unit or market prices. SAP Ariba solutions set the default currency.

To specify a particular currency, see Currency [page 45].

Advanced Pricing Details and Fields

Buyers and suppliers can specify pricing details in CIF and cXML catalogs, and buyers can specify pricing fields when creating CIF or cXML catalogs.

Advanced pricing details are sometimes referred to as `PriceBasisQuantity` or PBQ.

Advanced Pricing Details

Buyers and suppliers can specify different units of measure and pricing quantities in CIF and cXML catalogs, and SAP Ariba processes them using enhanced pricing calculation in requisitions, purchase orders, order confirmations, change orders, order confirmations, ship notices, invoices, and non-PO invoices. BMEcat catalogs do not support advanced pricing details.

SAP Ariba supports the following advanced pricing details:

- Quantity Based Pricing: Allows the unit price of an item to be based on a different price unit quantity than 1.
- Unit Conversion Pricing: Allows unit of measure conversion in the pricing calculation, when the unit of measure on the order differs from the pricing unit of measure.

Example: A buyer orders 20 reams of paper, with the unit price of $5. The unit price of $5 is for 2 cartons of paper, each carton containing 5 reams of paper. The unit conversion between the order unit (reams) and the pricing unit (cartons) is 0.2 (5 reams = 1 carton or 1 ream = 0.2 carton). The price is calculated using the unit conversion and the price unit quantity as:

Order Quantity	20
Order Unit	Reams
Unit Price	$5
Price Unit	Cartons
Price Unit Quantity	2
Unit Conversion	0.2

Price = Quantity * (Unit Conversion/Price Unit Quantity) * Unit Price

= 20* (0.2/2)*5 = $10

To specify advanced pricing details for the catalog items, add the advanced pricing fields to the CIF and cXML catalog files. These fields are used only when the commodities and supplier part numbers are ordered by quantity.

Purchasing users or contract agents can use catalogs containing the advanced pricing details when creating a requisition or contract.These fields are also available when suppliers use PunchOut catalogs. Buyers can view these fields in their SAP Ariba solutions after adding both customer catalog items and non-catalog items to a shopping cart.

Advanced Pricing Fields

While creating CIF or cXML catalogs, buyers can specify quantity based pricing and unit conversion pricing by adding the advanced pricing details fields to the catalog file. BMEcat catalogs do not spport advanced pricing details.

The following advanced pricing fields may be specified:

Field Name	Description
Price Unit Quantity	The quantity corresponding to the unit price of the item specified by the supplier. The unit price is based on the price unit quantity of the item. For example, the price of milk is $5 for 2 gallons. Here “2” in “2 gallons” is the price unit quantity.
Price Unit	The unit of measure for the unit price.
Unit Conversion	Ratio used to convert the ordered unit to the price unit. The unit price is based on the price unit quantity and the price unit. SAP Ariba uses the unit conversion to calculate the subtotal of an item using the following formula: Subtotal = Quantity * (Unit Conversion/Price Unit Quantity) * Unit Price
Price Unit Quantity Description	Any information for advanced pricing details. Use this field to store additional information on the unit conversion calculation. Note: This field name in the cXML catalog template must be Description.

The following sample shows a customer catalog item with advanced pricing details:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.024/cXML.dtd">
<Index>
    <SupplierID domain="DUNS">1000275</SupplierID>
    <Comments>
        Updated Sample1.xml for testing
    </Comments>
    <SearchGroup>
        <Name xml:lang="en">DVD</Name>
        SearchAttribute name="Type" type="string"/>
        <SearchAttribute name="Director" type="string"/>
        <SearchAttribute name="Region" type="string"/>
        <SearchAttribute name="Deleted Scenes" type="string"/>
        <SearchAttribute name="Length" type="string"/>
    </SearchGroup>
    <IndexItem>
        <IndexItemAdd>
            <ItemID>
                <SupplierPartID>17334</SupplierPartID>
            </ItemID>
            <ItemDetail>
                <UnitPrice>
                    <Money currency="USD">104.99</Money>
                </UnitPrice>
                <Description xml:lang="en"><ShortName>The X-Files: Season Two</
ShortName>Now you can own
                        the entire second season of The X-Files. All 25 classic
episodes are available for the
                        first time in this exclusive collector's edition. </
Description>
                <UnitOfMeasure>EA</UnitOfMeasure>
                <PriceBasisQuantity quantity="2" conversionFactor="0.5">
                        <UnitOfMeasure>BOX</UnitOfMeasure>
                        <Description xml:lang="en">1 Box is 2 EA and the unit price
is for 2
                            Boxes</Description>
                </PriceBasisQuantity>
                <Classification domain="UNSPSC_V13.5">55111514</Classification>
                <ManufacturerPartID>17334</ManufacturerPartID>
                <ManufacturerName>20th Century Fox</ManufacturerName>
                <URL>http://www.express.com</URL>
```

```
            </ItemDetail>
. . .
```

For more information about advanced pricing details for suppliers, see Order View Settings and Options in Receiving and Fulfilling Orders.

Advanced Pricing Requirements

- Buyers and suppliers must use cXML 1.0.24 or later to enable advanced pricing support. cXML version 1.0.24 supports the `PriceBasisQuantity` element.
- For buyers to use advanced pricing details, suppliers must upload a customer catalog containing items with the advanced pricing details. Customer catalog managers must download the customer catalog containing the line items with advanced pricing details and purchasing users or contract agents can use these catalogs to create requisitions or contracts. They can also specify advanced pricing details for non-catalog items.
- Buyers set transaction rules in their SAP Business Network accounts to allow suppliers to add or edit advanced pricing details on an order confirmation or invoice. For more information, see Transaction Rules for Advanced Pricing Fields.

Price Flags and Contract Files

Price flags in BMEcat catalogs indicate whether duty, insurance, freight, or packing are included in the price of an item. CIF and cXML catalogs do not support price flags. Contract files contain prearranged prices that buying organizations offer to specific customers.

Price Flags

In SAP Ariba solutions, item details do not show details for flagged prices, but they do show a message indicating that the displayed price includes duty, insurance, freight, or packing as applicable.

SAP Ariba solutions support all four `PRICE_FLAG` types: `incl_freight`, `incl_packing`, `incl_assurrance`, and `incl_duty`.

Contract Files

To offer special contract pricing, use contract compliance features.

For more information about contracts in SAP Ariba solutions, see *Managing contract compliance*.

> **Note**
>
> SAP Ariba solutions support contract compliance with CIF and cXML catalogs, but not with BMEcat catalogs.

Date-Dependent and Tiered Pricing in BMEcat Catalogs

BMEcat catalogs can include date-dependent pricing, which specifies different prices on different dates, and tiered pricing, which specifies different prices for different ordered quantities. CIF and cXML catalogs do not support date-dependent or tiered pricing.

You can define different levels of pricing in CIF catalogs for items based on different quantities and dates (validity period). For more information see Date and Quantity-Dependent Price Configuration in CIF Catalogs [page 37].

Date-Dependent Pricing

BMEcat catalogs can include date-dependent pricing, which specifies different prices on different dates. CIF and cXML catalogs do not support date-dependent pricing.

`ARTICLE_PRICE_DETAILS` can include a `DATETIME` that defines a start and end date for the specified pricing. The following example defines special pricing in December up to Christmas:

```
<ARTICLE_PRICE_DETAILS>
    <DATETIME type="valid_start_date">
        <DATE>2015-12-01</DATE>
    </DATETIME>
    <DATETIME type="valid_end_date">
        <DATE>2015-12-24</DATE>
    </DATETIME>
    <ARTICLE_PRICE type="net_customer">
        <PRICE_AMOUNT>9.99</PRICE_AMOUNT>
        <PRICE_CURRENCY>USD</PRICE_CURRENCY>
    </ARTICLE_PRICE>
</ARTICLE_PRICE_DETAILS>
```

Date-dependent pricing is useful for offering seasonal or holiday discounts. For date-dependent prices, SAP Ariba solutions display the current valid price in item details based on the server timezone. End users do not see information indicating that the price is date-dependent or when it will no longer be valid.

> **Note**
>
> If a customer catalog includes date-dependent pricing, the end date must always be after the start date. If the price details for any item in the customer catalog includes an end date that is before the start date, the customer catalog fails validation and is not imported into SAP Ariba solutions.

> **Note**
>
> For contracted items, the contract price when available always takes precedence over the customer catalog price during ordering and invoicing.

Tiered Pricing

`ARTICLE_PRICE_DETAILS` for a BMEcat catalog item can include multiple `ARTICLE_PRICE`s, each with a different `LOWER_BOUND`. These multiple prices define the item's price tiers. The following example shows pricing details for an item that is $9.99 for orders of up to 9 units and $8.99 if ordered in quantities of 10 or more:

```
<ARTICLE_PRICE_DETAILS>
    <ARTICLE_PRICE price_type="net_customer">
        <PRICE_AMOUNT>9.99</PRICE_AMOUNT>
        <PRICE_CURRENCY>USD</PRICE_CURRENCY>
        <LOWER_BOUND>1</LOWER_BOUND>
    <ARTICLE_PRICE>
    <ARTICLE_PRICE price_type="net_customer">
        <PRICE_AMOUNT>8.99</PRICE_AMOUNT>
        <PRICE_CURRENCY>USD</PRICE_CURRENCY>
        <LOWER_BOUND>10</LOWER_BOUND>
    </ARTICLE_PRICE>
</ARTICLE_PRICE_DETAILS>
```

For tiered prices, SAP Ariba solutions display the current valid price, based on the quantity ordered by the user, in item details. Above the current price is a **Pricing Information** link; clicking on that link displays pricing details for all tiers. If a user updates the quantity so that it is in another price tier, SAP Ariba solutions update the price at the time the shopping cart is updated.

> **Note**
>
> For contracted items, the contract price when available always takes precedence over the customer catalog price during ordering and invoicing.

Date and Quantity-Dependent Price Configuration in CIF Catalogs

You can define different levels of pricing for items based on different quantities and dates (validity period). Every different combination of these values can also define a different price.

Date and Quantity-Dependent Price Configuration

The `PriceConfiguration` field allows you to define different prices as a hash table of ["key-value"] pairs. The syntax is `[{"key"="value";"key"="value";"key"="value" * };{ * }]`. For example:

```
[{"datetime"= [{"type"=valid_start_date;"date"=2015-01-01};
{"type"=valid_end_date;"date"=2020-02-28}]; "articleprice"=
[{"priceType"=net_customer;"priceamount"=222.22;"pricecurrency"=USD;"pricefactor"=1;
"lowerbound"="1"};
{"priceType"=net_customer;"priceamount"=200.22;"pricecurrency"=USD;"pricefactor"=1;"
lowerbound"="20"}]}]
```

In this example, the item price is $222.22 USD if the quantity of the item is between 1 and 19 and is $200.22 USD if the item quantity is 20 or more. These prices are valid for the period `2015-01-01` to `2020-02-28`.

> **Note**
>
> SAP Ariba solutions support the `PriceConfiguration` field only in sites configured for managing customer catalogs using the content management system (CMS).

The following table lists the various keys and their supported values to define multiple price configurations:

Key Name	Required/Optional	Notes
`datetime`		
`"type"=valid_start_date`	Optional	Indicates the start date of the validity period in `YYYY-MM-DD` format.
`"type"=valid_end_date`	Optional	Indicates the end date of the validity period in `YYYY-MM-DD` format. **Note** The end date must always be after the start date. If the price details for any item in the catalog includes an end date that is before the start date, the customer catalog fails validation and is not imported into SAP Ariba solutions.
`articleprice`		
`priceType`	Required	Indicates the type of the price. SAP Ariba solutions support `net_customer` as the value that indicates that the price is customer-specific and excludes sales taxes.
`priceamount`	Required	The amount of the price.
`pricecurrency`	Optional	The currency of the price.
`pricefactor`	Optional	The (discount) factor that is multiplied by the `priceamount` value to determine the end price.
`lowerbound`	Optional	The lower limit of the item quantity for which the price is applied.

If multiple date-dependent prices are specified, only the price that is applicable on the date that the customer catalog is published is displayed in the customer catalog search results and item details pages. However, when users add the item to the shopping cart, the price that is valid on that date is updated. If a valid price is not available for the selected date, then users cannot add the item to the cart.

Lookup Key-Dependent Price Configuration in CIF Catalogs

The `PriceConfiguration` field allows you to define different prices for a CIF catalog item based on different values of a mapped entity configured as a price lookupkey.

Lookup Key-Dependent Price Configuration

You can define different prices for a CIF catalog item based on different values of a mapped entity configured as a price lookup key. The price of the item displayed in the catalog search results and item details pages is determined dynamically based on the business scenario and the value of the mapped price lookup entity. For example, if you define different prices for different regions, the price of the item to be displayed is determined by the region that the user is associated with.

Catalog managers can define different fields as the price lookup entities for the following different scenarios where catalog search is performed:

Scenario/Context	Description
Customer catalog home page search (outside the context of a requisition)	User performs catalog search directly from the **Catalog** dashboard and there are no items in the cart. If the cart contains items that had been added previously but not checked out, then the context is a requisition scenario.
Requisition	User in the process of creating a requisition from the **Home** dashboard clicks **Create** > **Requisition** and navigates to the customer catalog home page. User performs customer catalog search directly from the **Catalog** dashboard and there are one or more items in the cart that had been added previously but not checked out.

> **Note**
>
> The functionality provided by this feature is applicable only to the requisition and customer catalog home page search scenarios.

The `priceKey` attribute of the **PriceConfiguration** CIF field allows you to define the mapped entity as the price lookup key. The following is an example of a **PriceConfiguration** field value with the `priceKey` attribute:

```
[{"articleprice"=[{"priceType"=net_customer;"priceKey"="3000";"priceamount"=800;"pri
cecurrency"=USD;"pricefactor"=1};
{"priceType"=net_customer;"priceKey"="4000";"priceamount"=320;"pricecurrency"=USD;"p
ricefactor"=1};
{"priceType"=net_customer;"priceKey"="";"priceamount"=350;"pricecurrency"=USD;"price
factor"=1}]}]
```

In this example, the price of the item is $800 USD if the value of the mapped price lookup entity (e.g. Region Code) is 3000 and is $320 USD if the value of the entity (e.g. Region Code) is 4000. For all other values of the mapped price lookup entity (e.g. Region Code), the price of the item is $350 USD.

> **Note**
>
> SAP Ariba solutions support the `PriceConfiguration` field only in sites configured for managing customer catalogs using the content management system (CMS).

The following table lists the various keys and their supported values to define multiple price configurations:

Key Name	Required/Optional	Notes
`articleprice`		
`priceType`	Required	Indicates the type of the price. SAP Ariba solutions support `net_customer` as the value that indicates that the price is customer-specific and excludes sales taxes.
`priceKey`	Required	Indicates the mapped entity that is to be used as the price lookup key If you do not specify a value for the `priceKey` but specify a value for `priceamount`, then the price specified is applicable for all other values of the mapped price lookup entity unless specified.
`priceamount`	Required	The amount of the price.
`pricecurrency`	Optional	The currency of the price.
`pricefactor`	Optional	The (discount) factor that is multiplied by the `priceamount` value to determine the end price.

Use Case Scenario

Consider a business scenario where users procure items for **Region A** and **Region B** and the price of the catalog item is dependent on the region that it is being procured for.

Catalog managers can define two different prices for an item; one for **Region A** and the other for **Region B**, and configure the **Region** field as the price lookup entity. When a user in the process of procuring items for **Region A** performs a customer catalog search, the price of the item defined for **Region A** is fetched dynamically and displayed.

When creating price lookup entities, SAP Ariba recommends that customer catalog managers define the same field as the price lookup entity for both the User and Requisition classes. This is to ensure that the same price is displayed to users when they search for items outside the context of a requisition or from within the context of a requisition.

Defining different prices for an item for different regions and configuring the Region field as the mapped price lookup entity involves the following high-level steps:

1. The customer catalog manager navigates to **Catalog Manager** > **Catalog Views** > **Add Price Lookup Key** and configures the **Region** field in the User and Requisition classes as the price lookup key.

2. The catalog manager then creates and loads a catalog with different prices for items defined in the `PriceConfiguration` field.
   Consider the following `PriceConfiguration` field value for this example:

   ```
   [{"articleprice"=[{"priceType"=net_customer;"priceKey"="Region
   A";"priceamount"=800;"pricecurrency"=USD;"pricefactor"=1};
   {"priceType"=net_customer;"priceKey"="Region
   B";"priceamount"=320;"pricecurrency"=USD;"pricefactor"=1}]}]
   ```

   In this example, the price of the item is $800 USD if the Region value is Region A and is $320 USD if the region is Region B.
3. A user in the process of procuring items for Region A searches the customer catalog for the specific item.
4. Since the user is procuring for Region A, the SAP Ariba solution displays $800 USD as the price of the item.

Creating Price Lookup Keys

Use this procedure to define the fields available in the User and Requisition classes as price lookup keys. SAP Ariba recommends that you define the same field as the price lookup key for both the classes.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Select **Catalog Manager** > **Catalog Views**.
3. Click the **Add Price Lookup Key** tab.
4. Click **Add Price Lookup Key** to add a new price lookup key.
5. Select the class that you want the price lookup to be based on. For example, select **Requisition** if you want to create a price lookup key based on the fields available in the requisition class.

   The **Add Price Lookup Key** page displays the list of available fields that can be selected as the price lookup entities. Custom fields that are configured to be selected as the price lookup entities are also displayed.
6. Identify the field that you want to specify as the price lookup entity and click **Select**.
7. Click **Add Price Lookup Key** to add the field as the price lookup key.

Order Quantities for Customer Catalog Items

BMEcat and CIF catalog items can include minimum order quantities and quantity intervals. cXML catalogs do not support minimum order quantities or quantity intervals.

Order Quantities in BMEcat Catalogs

BMEcat catalogs support items that have a minimum order quantity or a quantity interval, and items that have both a minimum quantity and a quantity interval. For example, if an item has a minimum quantity of 10 and a quantity interval of 5, users can order 15 or 20 but cannot order 5 or 12. The following example shows the quantity definitions for this item:

```
<ARTICLE_ORDER_DETAILS>
    <ORDER_UNIT>1</ORDER_UNIT>
    <QUANTITY_MIN>10</QUANTITY_MIN>
    <QUANTITY_INTERVAL>5</QUANTITY_INTERVAL>
</ARTICLE_ORDER_DETAILS>
```

If users attempt to order less than the minimum quantity, or a quantity that does not match the interval, they are unable to add the item to a shopping cart and SAP Ariba solutions display an error message informing them of the issue so that they can correct the quantity.

Order Quantities in CIF Catalogs

You can specify the minimum order quantity and quantity interval for an item. If an item has a minimum order quantity, users cannot order less than that quantity. If an item has a quantity interval, the quantity that users order must match the interval. For example, if an item has a quantity interval of 2, users can order it in quantities of 2, 4, 6, 8, and so forth. Items can have both a minimum order quantity and a quantity interval; for example, users might need to order at least 2 of an item, and only be able to order it in quantities of 2, 4, 6, 8, and so forth.

You use the `MinimumQuantity` and `QuantityInterval` fields to specify the minimum order quantity and quantity interval numbers for an item.

> **Note**
>
> SAP Ariba solutions support the `MinimumQuantity` and `QuantityInterval` fields only in sites configured for managing customer catalogs using the content management system (CMS).

Content Unit and Packing Quantity

You can also specify the content unit of an item within an order unit and the packing quantity of the content units. For example, in a case of wine consisting of 12 bottles: Case is the order unit, Bottle indicates the content unit, and 12 the packing quantity.

In CIF catalogs, you use the `Content Unit` and `Packing Quantity` fields to specify content unit and packing quantity details.

In BMEcat catalogs, you use the `CONTENT_UNIT` and `NO_CU_PER_OU` elements to specify content units and packing quantity.

```
<ARTICLE_ORDER_DETAILS>
        <ORDER_UNIT>CS</ORDER_UNIT>
        <CONTENT_UNIT>BO</CONTENT_UNIT>
        <NO_CU_PER_OU>12</NO_CU_PER_OU>
        <PRICE_QUANTITY>1</PRICE_QUANTITY>
        <QUANTITY_MIN>2</QUANTITY_MIN>
        <QUANTITY_INTERVAL>2</QUANTITY_INTERVAL>
    </ARTICLE_ORDER_DETAILS>
```

> ### Note
>
> If any one of the field values for an item is missing, then neither of the field values are displayed in the user interface.
>
> The **Content Unit** and **Packing Quantity** fields are displayed in the catalog search results and the item details pages.

Related Information

Territory Available Field and Territory Codes for Customer Catalog Items [page 43]
Currency, Currency Codes, and Alternate Currency for Customer Catalogs [page 45]

Territory Available Field and Territory Codes for Customer Catalog Items

Use the `Territory Available` field and ISO 3166-1 or 3166-2 territory codes to display certain CIF or cXML catalog items to specific users based on geography. Users who reside outside of the countries or regions listed cannot see that item. Customer catalog items with no `Territory Available` field appear to all users.

> ### Note
>
> Although the BMEcat standard includes a `TERRITORY` field in `ARTICLE_PRICE`, SAP Ariba does not currently support that field, and SAP Ariba solutions ignore it during customer catalog import.

CIF example:

```
"US,GB"
```

cXML example:

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

> **Note**
>
> SAP Ariba solutions ignore the `Territory Available` field by default. However, customer catalogs in the SAP Ariba Catalog solution can be filtered by specifying `Territory Available` as a constraint in the `CustomFilterQuery` extrinsic data. Contact your customers to find out whether they have enabled this functionality in their SAP Ariba Catalog solution.

Do not use these ISO codes to specify currency or language. For more information, see Currency [page 45] and Catalog Language [page 147].

> **Note**
>
> SAP Ariba solutions do not support ISO 3166-2 territory codes.

ISO 3166-1 Country Codes

Buying organizations typically 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.

For example:

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

Alpha-3: `USA, GBR, FRA`

For more information about Alpha-2 codes and to look up specific codes, visit www.unece.org .

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`

> **Note**
>
> SAP Ariba solutions do not support ISO 3166-2 territory codes.

For more information about ISO3166-2 codes visit codes and to look up specific codes, visit www.unece.org .

Currency, Currency Codes, and Alternate Currency for Customer Catalogs

You can specify currency, currency codes, or alternate currency in catalogs to best suite your customer's currency type.

Currency

With CIF catalogs, you can specify currency either at the customer catalog header level or the customer catalog item level. This lets you specify default currency at the customer catalog header level and have individual customer catalog items that override that default.

With cXML catalogs, currency is specified at the item level only.

With BMEcat catalogs, currency can be specified at the header or item level. If currency is specified at the header level, that currency applies to all items that do not have currency defined at the line level. If the item-level currency differs from the header-level currency, it overrides the header-level currency for that item.

If a catalog specifies no currency, it uses the system currency specified within the SAP Ariba solution.

Currency Codes

Specify currency with three-letter ISO 4217 currency codes.

For example:

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

For more information about ISO 4217 currency codes visit codes and to look up specific codes, visit 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.

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

Related Information

Territory Available Field and Territory Codes for Customer Catalog Items [page 43]

Effective Date, Expiration Date, and Lead Time

For each CIF or cXML catalog item, you can specify the calendar period during which it is available and how long it takes to deliver. BMEcat catalogs do not support effective date, expiration date, or lead time.

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 to avoid validation errors. If you do not need to specify the start of availability, leave out `Effective Date`. Similarly, if you do not need to specify the end of availability, leave out `Expiration Date`. If you use both fields, the value of `Expiration Date` must be a date after the value of `Effective Date` to avoid validation errors.

Effective Date

The `Effective Date` field indicates the start of product or service availability. SAP Ariba solutions do not display the item in customer catalog search results before the effective date.

Expiration Date

When an item expires, SAP Ariba solutions no longer display it to users. In-progress purchase requisitions, however, continue to list the item. Use `Expiration Date` to limit the duration of negotiated pricing.

You can also use `Expiration Date` for items that will be superseded by newer items in the future. Neither SAP Business Network nor SAP Ariba solutions issue notifications when items expire.

SAP Ariba solutions hide entire expired line items, not just their prices.

Lead Time

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

The `Lead Time` value includes all tasks performed by the supplier, logistics providers, and delivery services to deliver the item, including the internal tasks of picking, packing, and shipping. Use Lead Time = 1 to indicate same-day delivery. Don't use Lead Time = 0.

SAP Ariba solutions do not allow users to set a need-by date in purchase orders that precedes `Lead Time`.

Supplier and Manufacturer URLs for Customer Catalog Items

Each CIF or cXML catalog item can have a `Supplier URL` and a `Manufacturer URL` that allow users to view information about that item through the World Wide Web.

BMEcat catalog items can have URLs as related documents, including supplier or manufacturer URLs. However, they do not have explicit supplier or manufacturer URL fields. For details, see Related Documents [page 57].

> ### Note
>
> These URLs are not used for PunchOut catalogs.

If either of these URLs are present for an item, SAP Ariba Procurement solutions display the URLs as hyperlinks. Clicking it displays a new web browser window containing the specified website.

The Supplier and Manufacturer URLs are displayed in the customer catalog item details pages. Use these URLs to provide information that helps customers understand your products or services. For example, you can display:

- Pictures and datasheets for items (see Images [page 68])
- 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

  > ### Note
  >
  > To access supplier or manufacturer websites, SAP Ariba solution users must have Internet access from their desktop computers. Not all buying organizations enable users to access the Internet, ask your customers whether their users have Internet access.

Related Information

Optional Supplemental Attributes for Customer Catalog Fields [page 47]
Effective Date, Expiration Date, and Lead Time [page 46]

Optional Supplemental Attributes for Customer Catalog Fields

Supplemental attributes are optional CIF or cXML catalog fields that help customers 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.

Suppliers can add supplemental attributes to their customer catalogs, if appropriate. Customers can configure their systems to recognize and take advantage of this additional data. SAP Ariba solutions can be configured

to recognize and use supplemental attributes. For SAP Ariba solutions, use type definition files to declare supplemental attribute fields. For more information, see Type definition files [page 156].

Supplemental attribute names must begin with a letter and can include letters, digits, periods (.), dashes (-), and underscores (_). They cannot contain spaces. The recommended maximum length of a supplemental attribute name combined with its value is 75 characters.

For naming restrictions in SAP Ariba solutions, see Parametric and supplemental attribute naming restrictions [page 157].

Related Information

Supplier and Manufacturer URLs for Customer Catalog Items [page 47]

Parametric Search Data

SAP Ariba solutions support optional parametric search data that enables users to perform sophisticated searching based on product-specific properties. Parametric searching enables users to find items they need within large customer catalogs of similar products.

SAP Ariba Buying applications do not support parametric searching. To enable parametric searching, add parametric data to items in your CIF or cXML catalogs.

Overview of Parametric Searching

Include parametric data in your customer catalog file to enable SAP Ariba solutions to display new fields in the product search window. You define parametric data as lists of "key-value" pairs.

For example, if you supply pipes, you can add parametric data to enable users to search by length or diameter. Or, if you supply computer monitors, you can enable users to search by screen size or dot pitch. SAP Ariba solutions generate the contents of this page and its dropdown menus by aggregating the parametric data from your customer catalogs. You do not need to separately list all parametric data or enumerate all possible parametric attribute values.

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`.

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.
- Some examples of 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 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 ten 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.

To learn how to declare parametric data fields in an SAP Ariba solutions (cXML 1.2.008 and later) and for examples, see Type Definition Files [page 156].

Related Information

Parametric Attribute Names and Data Types [page 51]

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 customer catalogs.

For more information on parametric data, see Type Definition Files [page 156].

cXML Parametric Data Example

In cXML parametric data appears as a `SearchDataElement` nested in a `searchGroupData` element.

The following example shows parametric data in a cXML catalog.

```
...
<IndexItem>
    <IndexItemAdd>
        item ID
        <ItemDetail>
            item description
        </ItemDetail>
        <IndexItemDetail>
            <SearchGroupData searchGroupName="Casings">
                <SearchDataElement name="TYPE" value="SEAMLESS"/>
                <SearchDataElement name="SIZE" value='20"'/>
                <SearchDataElement name="WEIGHT PER FOOT" value=”94.00”/>
            </SearchGroupData>
        </IndexItemDetail>
    </IndexItemAdd>
</IndexItem>
```

For information about specifying parametric data in cXML, see Topics about cXML catalogs [page 105] and Type Definition Files [page 156].

CIF Parametric Data Example

Parametric data shows inside curly braces ({}) in a CIF catalog.

For 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: 2005-01-18 00:00:00
ITEMCOUNT: 3
DATA
599238,66056062,66056062,"20"" Carbon Steel Seamless Casing",99998050,57.95,EA,1,CAS
INGSRUS,,,77.95,
```

```
{SIZE=20\";"WEIGHT PER FOOT"=94.00;THICKNESS=0.438;THREADTYPE="BUTTRESS
THREAD";},Casings
599238,66056055,66056055,"20"" Carbon Steel Seamless Casing",99998050,58.95,EA,1,CAS
INGSRUS,,,78.95,
{SIZE=20\";"WEIGHT PER FOOT"=133.00;THICKNESS=0.635;THREADTYPE="BUTTRESS
THREAD";},Casings
599238,66056048,66056048,"20"" Carbon Steel Seamless Casing",99998050,59.95,EA,1,CAS
INGSRUS,,,79.95,
{SIZE=20\";"WEIGHT PER FOOT"=137.00;THICKNESS=0.750;THREADTYPE="BUTTRESS
THREAD";},Casings
ENDOFDATA
```

The value `20\"` is used because the inch symbol (") must have a backslash (\) in front of it. Also, be sure to use a semicolon (;) after each parametric value.

For information about specifying parametric data in CIF, see Special field syntax rules [page 95].

Parametric Attribute Names and Data Types

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 customer 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, Coarse
MATERIAL: Cloth, Vinyl, Leather	

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

> **Note**
>
> Ariba Buyer 6.0 displays `Parametric Name` in mixed case. Ariba Buyer 7.0 and later display `Parametric Name` in all uppercase. In Ariba Buyer 7.1 and earlier, parametric attribute values are case-insensitive; it changes all characters to uppercase.

Parametric Name Data Aggregation

SAP Ariba Buying solutions aggregate `Parametric Name` for all catalogs from all suppliers. This aggregation lets users search across all suppliers with a single query.

> **Note**
>
> Ariba Buyer 7.1 and earlier aggregate `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 do not aggregate parametric data from different suppliers.

Parametric Data Types

In CIF, the parametric data type is always `string`. An example of a `Parametric Name` value is "COLOR". 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 Overview of Type Definition Files [page 156].

Parametric Attribute Name and Value Requirements

Parametric attribute names and values can use any character set listed in Catalog Character Set and Character Encoding [page 153].

Maximum Length	In CIF, each hash table (parametric attribute names and values) can contain up to 255 characters.)
First Character	Must be a letter. For the definition of "letter," see www.w3.org/TR/REC-xml#NT-Letter .
Legal Characters	Letters, digits, spaces, periods (.), dashes (-), and underscores (_).

SAP Ariba Buying solutions remove leading and trailing spaces. Then, it replaces all punctuation (except dashes) with underscores (_).

Use the same attribute names across parametric types to maintain consistency for users. For example, use the attribute name `length` for all parametric types instead of `length` for one and `diameter` for another.

For SAP Ariba Buying solutions naming restrictions, see Parametric and supplemental attribute naming restrictions [page 157].

Parametric attribute names are case-insensitive. SAP Ariba Buying solutions use a separate, dedicated, user-visible value for each attribute name. Parametric attribute values are case-insensitive by default, but can be configured to be case-sensitive using the `isCaseSensitive` property.

Required for Ordering Attribute

To increase order accuracy and obtain information from the user when an item is ordered, use the `isRequiredForOrdering` type attribute property in a type definition file. When this attribute is specified, the user must fill out the field value before submitting the purchase order.

A customer catalog file imported with no value on a `isRequiredForOrdering` field passes validation as long as the field is not a standard required field. Any values provided in the customer catalog for the field set with `isRequiredForOrdering` appear as editable fields when the user adds the item to a requisition. SAP Ariba Buying and Invoicing does not support the `isRequiredForOrdering` attribute.

The `isRequiredForOrdering` attribute also includes the value of that field on the purchase order that is sent to the supplier. Just because a customer catalog item is a partial item does not mean the information the user specifies will be sent to the supplier; the `isRequiredForOrdering` attribute must also be specified for the fields.

For more details about type definition files and type attribute properties see Type Attributes [page 160].

Relevance Ranking for Customer Catalog Items

Relevance ranking determines how search results are ordered in procurement applications. Use the relevance fields only when instructed to by your buyer customer.

Several factors can determine the relevance that procurement applications use to map into a single ordering:

- The relevance given to the item in the CIF or cXML catalog file (using the `defaultrelevance` attribute)
- Keyword matching for the item in BMEcat (using the `KEYWORD` attribute) and CIF catalog item
- Whether the supplier has been designated a preferred supplier
- The popularity of the item, based on how often it has been purchased in the past
- The position of the item in the customer catalog file
- The free text relevance score computed by the SAP Ariba solution's customer catalog search engine

Relevance is determined by defining a hierarchy for these factors and ranking each factor. For example, when a procurement application considers the relative relevance of two items, the first factor determines which one has higher relevance. If both items have the same value or no value for that factor, then the next factor is considered, and so on.

The following criteria determines which factor is most important and which is least important:

1. The most important factor is free text relevance as determined by the customer catalog search engine. Typically the free text search involves only a single word or the ANDing of multiple words. In these cases, relevance is based on how closely the words in the document match the search terms. For example, an exact match gets high relevance. If it matches only because of soundex or fuzzy spelling, then the relevance is low. Matching because of wildcarding receives medium relevance. The number of words that match in the case of free text ORing is also very important. If there is no free text term on the query, all results have the same relevance from the customer catalog search engine.
2. In the CIF or cXML catalog file, the administrator or supplier can specify a relevance for each item by using a value between 0 and 1000, with 0 being the least relevant. This attribute in the catalog file is called `defaultrelevance`. For an example, see Relevance Ranking Example [page 101].
3. A Boolean attribute in the CIF or cXML catalog can indicate whether a supplier is preferred. Items from preferred suppliers appear before items from other suppliers.

4. Popularity of the item among all users in the realm is considered. This is a function of how often the item is ordered and added to favorites by users at the same company.
5. The line number in the file in which the items appear. Appearing early in the file provides higher relevance than appearing later in the file.

In addition to relevance, a second mechanism allows CIF or cXML catalog owners to highlight a small set of rows. Currently this is used in user-based and PunchOut-based ordering. For example, during a search a few items are collected that the user has purchased recently or which she added to her favorites. She also collects a few items that are store-level or aisle-level PunchOut items. Even though these items might not be the most relevant to the user's query, if they satisfy the search criteria then they should be displayed because they are either relevant to the user or, as PunchOut items, they represent many other items.

This mechanism can also be used to gather the most relevant rows from a preferred supplier and highlight them in a search result.

Partial Items, Required Fields and IsPartial Field, and Enumeration Values

Your SAP Ariba Buying solutions support the use of partial catalog items in cXML and CIF catalogs. Partial items are regular customer catalog items imported into Ariba Administrator with certain fields deliberately missing. BMEcat catalogs do not support partial items.

Using partial items in your catalogs lets customers search for and find items that are not fully specified customer catalog items.

Partial items also allow customers to order services and other customizable items that might contain information that the customer must supply when the item is ordered. When a partial item is added to the requisition, users are prompted to fill in the missing fields, which converts the partially specified items into fully specified items on the order. The use of partial items for service items increases order accuracy, supports more specific ordering, and reduces the number of customer catalog items.

Required Fields and IsPartial Field

Some customer catalog fields are required fields and generate errors if they are missing when customers import the catalog into Ariba Administrator. Using the `IsPartial` field lets a partially specified items pass validation when imported.

To enable the use of partial items in CIF catalogs, the `IsPartial` field must be included in the `FIELDNAMES` header of the CIF catalog file and then must be specified for each partial item in the body of the customer catalog.

For example:

```
CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
COMMENTS: This is a sample CIF 3.0 file for loading a partial item
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,
      IsPartial
     , ContractId, Parametric Name, Parametric Data
TIMESTAMP: 2005-03-29
UNUOM: TRUE
ITEMCOUNT: 1
DATA
010010010,PartialItem 1.1,,Partial Item test case 1.1,26111701,,,,Mantell Lawn
Inc.,,,,true,1234,System:LawnService,
{frequency=once;weeding=true;planttrimming=true;}
ENDOFDATA
```

In cXML, partial items are declared with the extrinsic tag. See Glossary of Catalog Terms [page 11] to learn more about extrinsics. You can then create catalogs containing items with pieces of data missing, as long as `IsPartial` is set to `true` for that item. By default, `IsPartial` is set to `false` for both CIF and cXML catalogs.

You only need to use `IsPartial` to declare that an item is partial if one or more required customer catalog fields do not contain values. However, the following mandatory fields cannot be missing even if the `IsPartial` field is used: `Classification Code`, `Description`, and `Supplier Part ID`. You can however, omit the following fields: `Price`, and `UOM`.

> **Note**
>
> You need to have a classification whether it comes from `Classification Code` or the `SPSC` field.

Enumeration Values

Enumeration values are all possible values allowed for a specific item.

For example, if you have a customer catalog containing t-shirt items that come in different colors, specify the color `TypeAttribute` of the `TSHIRT` type in the type definition file, then list all possible color values (red, blue, black, white, yellow) and set `isRequiredForOrdering` to `yes`. Then in the customer catalog file, for each t-shirt item, specify the color values allowed for that particular t-shirt (red, blue, black). Enumeration values are specified through the `TypeAttribute` element in type definition files. For more information about enumeration values and type definition files, see Type Attributes [page 160].

You can also use enumeration values to increase the order accuracy of partial items by displaying the allowable values for a specific field. When enumeration values are implemented on a required field, SAP Ariba Buying solutions display a dropdown menu containing values that the user must choose from when ordering an item.

Related Items and Documents in Customer Catalogs

In BMEcat, CIF, and Excel (.xlsx) catalogs, related items can be mandatory items, accessories, spare parts, similar items, and follow-up items. BMEcat, CIF, and Excel catalog items can also include related documents such as user manuals, product specifications, or links to external information. cXML catalogs do not support related items or related links.

Related Item in BMEcat Catalogs

`ARTICLE`s can include `ARTICLE_REFERENCE` elements with the following types:

- `similar`, for items that are similar or equivalent to the item
- `accessories`, for optional accessories for the item
- `sparepart`, for optional spare parts for the item
- `followup`, for more advanced versions of the item
- `mandatory`, for items that must be ordered together with the item

For related items, SAP Ariba solutions only support using `ART_ID_TO` to identify the item. The following example shows a mandatory item:

```
<ARTICLE_REFERENCE type="mandatory">
    <ART_ID_TO>A-123</ART_ID_TO>
</ARTICLE_REFERENCE>
```

SAP Ariba solutions display similar and follow-up items, accessories, and spare parts in different sections of the main item's **Item Details** page. Users must navigate to those items and add them to the shopping cart separately. If a user adds a main item with mandatory items to the shopping cart, SAP Ariba solutions automatically add the mandatory items as well. If a user then deletes the main item from the shopping cart, SAP Ariba solutions delete the mandatory items after showing the user a message and asking for confirmation of the deletion.

Related Items in CIF Catalogs

SAP Ariba solutions allow you to associate other items as accessories, spare parts, follow-up, or similar items to the main items. You can use the `RelatedItems` field to define this association. You can specify multiple related items as a hash table of ["key-value"] pairs. The syntax is `[{"key"="value";"key"="value"};{ * }]`. The string within braces ({}) indicates one item. For example:

```
[{"type"="accessories";"artidto"="z2_tripod"};
{"type"="mandatory";"artidto"="z2_case"}; {"type"="followup";"artidto"="z3_case"}]
```

The following table lists the various keys and their supported values to define related items:

Key Name	Required/Optional	Notes
`type`	Required	Indicates the type of the item in relation to the main item. SAP Ariba solutions support the following values for related items: • `mandatory` • `similar` • `sparepart` • `accessories` • `followup`

Key Name	Required/Optional	Notes
`artidto`	Required	Used to identify the related item. All related items must be in the same catalog. SAP Ariba solutions do not support related items from different customer catalogs.

Related Documents in BMEcat Catalogs

`MIME_INFO` for a BMEcat catalog item can contain multiple `MIME` fields with different types to define related documents. SAP Ariba solutions support the following mime types for documents:

- `application/pdf`
- `url`
- `text/html`
- `text/plain`
- `application/xlsx`
- `application/docx`

The following example defines two different related documents for an item, a user manual and a website URL where product specifications are displayed:

```
<MIME_INFO>
    <MIME>
        <MIME_TYPE>application/pdf<MIME_TYPE>
        <MIME_SOURCE>http://12.34.56.789:8888/manuals/user_manual.pdf</MIME_SOURCE>
        <MIME_DESCR>User Manual</MIME_DESCR>
        <MIME_ALT>PDF user manual for this item</MIME_ALT>
        <MIME_PURPOSE>data_sheet</MIME_PURPOSE>
        <MIME_ORDER>5</MIME_ORDER>
    </MIME>
    <MIME>
        <MIME_TYPE>url</MIME_TYPE>
        <MIME_SOURCE>www.thismanufacturer.com/productinfo/
model123_specifications.html</MIME_SOURCE>
        <MIME_DESCR>Product Specifications</MIME_DESCR>
        <MIME_ALT>Manufacturer's specifications for this item</MIME_ALT>
        <MIME_PURPOSE>data_sheet</MIME_PURPOSE>
        <MIME_ORDER>6</MIME_ORDER>
    </MIME>
</MIME_INFO>
```

> **Note**
>
> Although the BMEcat DTD does not require `MIME_PURPOSE`, SAP Ariba solutions use it to properly display `MIME`s in the correct area of catalog item details. For related documents, always use `data_sheet` so that they display in the **Related Documents** area of item details. SAP Ariba solutions ignore all items that do not have a `MIME_PURPOSE` during catalog import.
>
> SAP Ariba solutions enforce a character limit of 250 each for `MIME_SOURCE` and `MIME_DESCR` and of 50 for `MIME_ALT`.

Related Documents in CIF Catalogs

You can attach documents related to the catalog item to be displayed in the **Item Details** pages using the `Attachments` field. You can specify multiple related documents as a hash table of ["key-value"] pairs. The syntax is `[{"key"="value";"key"="value";"key"="value" * };{ * }]`. The string within braces ({}) indicates one document.

For example:

```
[{"mimetype"="application/pdf";"mimesource"="http://10.138.169.202:8888/myapp/
nexus_guidebook.pdf";"mimedescr"="User Guide for Xperia
Users";"mimealt"="Xperia User Guide";"mimepurpose"="data_sheet";"mimeorder"="1"};
{"mimetype"="application/pdf";"mimesource"="http://10.138.169.202:8888/myapp/
nexus_manual.pdf";"mimedescr"="User Guide for Xperia Users";"mimealt"="CIFXperia
User Guide";"mimepurpose"="data_sheet";"mimeorder"="2"}]
```

The following table lists the various keys and their supported values to define attachments:

Key Name	Required/Optional	Notes
`mimetype`	Optional	Indicates the type of the document. SAP Ariba solutions support the following values for related documents: • `application/pdf` • `url` • `text/html` • `text/plain` • `application/xlsx` • `application/docx`
`mimesource`	Required	Indicates the source of the document. The value can be a URL or the pathname to a document.
`mimedescr`	Optional	Indicates the description of the document.
`mimealt`	Optional	Indicates the alternative text for the document.
`mimepurpose`	Optional	Indicates the purpose of the document. solutions support `data_sheet` as the value for related documents.
`mimeorder`	Optional	Indicates the order in which the documents are to be displayed.

For information about related items and documents in Excel catalogs, see Customer Catalogs in Microsoft Excel (.XLSX) Format [page 139].

Related Information

Additional Item Details in BMEcat Catalogs [page 59]

Additional Item Details in BMEcat Catalogs

The following sections describe the additional article detail fields in BMEcat catalogs that are supported by SAP Ariba solutions.

European Article Number

The EAN element is used to define the European article number of a customer catalog item. The permitted value is a single string value.

This field value is mapped to the new **Additional Part IDs** customer catalog field by default. If SAP Ariba creates an extensible customer catalog field for the EAN value, it can be configured to be:

- displayed in the customer catalog search results and the item details pages
- available as an advanced search option for customer catalog search.

Buyer Article ID

The BUYER_AID type= element is used to specify the article numbers used by the buying organizations. The type attribute specifies the type of the ID. If your organization uses multiple types of IDs to classify an item, you can specify type of ID as a separate type in the BUYER_AID. The values for each type must be different.

In the following example, SAP_MATNR and PRD_ID indicate the different types of buyer article IDs:

```
<ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Motorola Micro USB Data Cable</DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Motorola Micro USB Data Cable-Compatible with
Motorola Droid RARZ, RAZR Maxx, Droid 3, Droid 4</DESCRIPTION_LONG>
                <BUYER_AID type="SAP_MATNR">CH.123</BUYER_AID>
               <BUYER_AID type="PRD_ID">657609897</BUYER_AID>
                <DELIVERY_TIME>14</DELIVERY_TIME>
                <KEYWORD>USB</KEYWORD>
                <KEYWORD>Cable</KEYWORD>
                <KEYWORD>Motorola</KEYWORD>
                <ARTICLE_STATUS type="core_article">true</ARTICLE_STATUS>
            </ARTICLE_DETAILS>
```

This field value is mapped to the new **Additional Part IDs** customer catalog field by default. If the **Additional Part IDs** is configured to be displayed in the customer catalog search results and the item details pages, then the buyer article ID type and the value are displayed. If you have multiple types specified, then all the types are displayed separated by commas. If you want the buyer ID types to be displayed as separate catalog fields, contact SAP Ariba Customer Support to create extensible customer catalog fields and to configure your site accordingly.

If SAP Ariba creates extensible customer catalog fields for each type of the BUYER_AID element, you can configure these fields to be available as an advanced search options for customer catalog search.

Special Treatment Class

The `SPECIAL_TREATMENT_CLASS type=` element is used to specify the classification of dangerous and hazardous goods. The `type` attribute specifies the dangerous goods classification scheme and the value indicates the actual class within this scheme. This field value is mapped to the new **SpecialTreatmentClass** catalog field by default.

In the following example, `GHS` and `GHSSignal` indicate the classification scheme.

```
<ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>SEALED LEAD-ACID BATTERY</DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Sealed Lead-Acid batteries for use in standby
applications.</DESCRIPTION_LONG>
                <EAN>LAB54894</EAN>
                <BUYER_AID type="SAP_MATNR">3658</BUYER_AID>
                <DELIVERY_TIME>21</DELIVERY_TIME>
                <SPECIAL_TREATMENT_CLASS type="GHS">H302</SPECIAL_TREATMENT_CLASS>
               <SPECIAL_TREATMENT_CLASS type="GHS">P280</SPECIAL_TREATMENT_CLASS>
               <SPECIAL_TREATMENT_CLASS type="GHSSignal">DANGER</
SPECIAL_TREATMENT_CLASS>
                <KEYWORD>Battery</KEYWORD>
                <KEYWORD>Lead-Acid</KEYWORD>
        </ARTICLE_DETAILS>
```

If you want the icons or pictograms that indicate the nature of the danger to be displayed for the items in the customer catalog search results and the item details pages, contact SAP Ariba Customer Support.

Article Status

The `ARTICLE_STATUS type=...` element is used to classify an article in terms of its special characteristics. The status type is specified by the `type` attribute and the value indicates the text description of the special characteristics. For example, `<ARTICLE_STATUS type="core_article">true</ARTICLE_STATUS>` indicates that the item is classified as a core article and `<ARTICLE_STATUS type="new_article">true</ARTICLE_STATUS>` indicates that the item has been recently added to the catalog.

This field value is mapped to the new **ItemSpecialStatus** customer catalog field by default.

You can configure your system to maps these fields to existing out of the box customer catalog fields. For example you can create a customer catalog rule to enrich a customer catalog item as a **Preferred Item** if the item's article status is classified as a core article.

Multiple Product Classification Systems in BMEcat Catalogs

SAP Ariba solutions provide support for multiple product classification systems and enable users to shop for items based on the classification systems. Users can browse the hierarchy of each classification system and search for relevant items.

SAP Ariba solutions process the information related to multiple classification codes specified in the BMEcat catalog files and generate various customer catalog hierarchies based on each classification system.

The **Classification System** option displayed in the **Shop By** section of the customer catalog home page enables users to browse the hierarchy of each classification system and search for relevant items. To enable this feature, have your Designated Support Contact (DSC) submit a Service Request (SR). An SAP Ariba Customer Support representative will follow up to complete the request.

Supporting multiple classification systems in BMEcat catalogs involves:

- defining the classification system and the hierarchical assignment of the constituent groups using the `CLASSIFICATION_SYSTEM` and `CLASSIFICATION_GROUPS` elements
- mapping items to the defined classification systems and feature groups using the elements within the `ARTICLE_FEATURES` element

Defining Classification Systems

The information related to multiple product classification systems is defined using the `CLASSIFICATION_SYSTEM` and `CLASSIFICATION_GROUPS` elements in BMEcat catalogs. SAP Ariba solutions process this information and generate customer catalog hierarchies based on each classification system.

Element	Mandatory	Type and length	Notes

`CLASSIFICATION_SYSTEM`	No		Used to define a product classification system.
			Example:
			`<CLASSIFICATION_SYSTEM>`
			`  ...`
			`  <CLASSIFICATION_GROUPS>`
			`  ...`
			`  <CLASSIFICATION_GROUP type="node">`
			`  ...`
			`  </CLASSIFICATION_GROUP>`
			`  <CLASSIFICATION_GROUP type="node">`
			`  ...`
			`  </CLASSIFICATION_GROUP>`
			`  ...`
			`  </CLASSIFICATION_GROUPS>`
			`</CLASSIFICATION_SYSTEM>`

Element	Mandatory	Type and length	Notes
CLASSIFICATION_SYSTEM_NAME	Yes	String (20)	Indicates the unique identifier of the classification system. It must combine the short name of the classification system and the version number so that it can be referenced from the ARTICLE_FEATURES element if required. The format for the identification number should follow the pattern: <Name>-<MajorVersion>.<MinorVersion> Example: ETIM-1.0, ECLASS-3.0, UNSPSC-3.0 <CLASSIFICATION_SYSTEM_NAME>ETIM-1.0</CLASSIFICATION_SYSTEM_NAME> A list of pre-defined names for standardized classification systems must also be defined using the REFERENCE_FEATURE_SYSTEM_NAME element.
CLASSIFICATION_SYSTEM_FULLNAME	No	String (60)	Indicates the full name of the classification system. Example: <CLASSIFICATION_SYSTEM_FULLNAME>Electrotechnical information model</CLASSIFICATION_SYSTEM_FULLNAME>

CLASSIFICATION_GROUPS	No		Specifies the group of the classification system as well as its hierarchical assignment.
			
			`<CLASSIFICATION_GROUPS>`
			`  ...`
			`  <CLASSIFICATION_GROUP type="node">`
			`  ...`
			`  </CLASSIFICATION_GROUP>`
			`  <CLASSIFICATION_GROUP type="node">`
			`  ...`
			`  </CLASSIFICATION_GROUP>`
			`  ...`
			`</CLASSIFICATION_GROUPS>`

CLASSIFICATION_GROUP type="…"			Describes a group including its features within the classification system. The type attribute specifies the position of the group within the hierarchy. The following values are allowed: • node - A group or subgroup of the classification which does not contain individual articles but only other subgroups. • leaf - A group on the lowest level of the classification which does not contain any subgroups. It contains articles only. Example: <CLASSIFICATION_GROUP type="node">

Element	Mandatory	Type and length	Notes
CLASSIFICATION_GROUP_ID	Yes	String (60)	Specifies the unique identification of the group. Example: <CLASSIFICATION_GROUP_ID>17000000</CLASSIFICATION_GROUP_ID>
CLASSIFICATION_GROUP_NAME	Yes	String (60)	Specifies the unique name of the group. Example: <CLASSIFICATION_GROUP_NAME>Cable and wire entry systems</CLASSIFICATION_GROUP_NAME>
CLASSIFICATION_GROUP_DESCR	No	String (250)	Specifies the description of the group.
CLASSIFICATION_GROUP_PARENT_ID			References the unique identification (CLASSIFICATION_GROUP_ID) of the parent group. This element must not be specified if there is no parent group for the group. Example: <CLASSIFICATION_GROUP_PARENT_ID>17010000</CLASSIFICATION_GROUP_PARENT_ID>

The following lists a BMEcat catalog that defines a classification system using the CLASSIFICATION_SYSTEM and CLASSIFICATION_GROUPS elements:

```
<CLASSIFICATION_SYSTEM>
    <CLASSIFICATION_SYSTEM_NAME>ETIM-1.0</CLASSIFICATION_SYSTEM_NAME>
    <CLASSIFICATION_SYSTEM_FULLNAME>Electrotechnical information model</
CLASSIFICATION_SYSTEM_FULLNAME>
    <CLASSIFICATION_SYSTEM_VERSION>1.0</CLASSIFICATION_SYSTEM_VERSION>
    <CLASSIFICATION_SYSTEM_DESCR> Description of the classification system.</
CLASSIFICATION_SYSTEM_DESCR>
    ...
    <CLASSIFICATION_GROUPS>
        <CLASSIFICATION_GROUP type="node">
            <CLASSIFICATION_GROUP_ID>17000000</CLASSIFICATION_GROUP_ID>
            <CLASSIFICATION_GROUP_NAME>Cable and wire entry systems</
CLASSIFICATION_GROUP_NAME>
        </CLASSIFICATION_GROUP>
        <CLASSIFICATION_GROUP type="node">
            <CLASSIFICATION_GROUP_ID>17010000</CLASSIFICATION_GROUP_ID>
            <CLASSIFICATION_GROUP_NAME>Cables</CLASSIFICATION_GROUP_NAME>
            <CLASSIFICATION_GROUP_PARENT_ID>17000000</
CLASSIFICATION_GROUP_PARENT_ID>
        </CLASSIFICATION_GROUP>
        <CLASSIFICATION_GROUP type="leaf">
            <CLASSIFICATION_GROUP_ID>17010100</CLASSIFICATION_GROUP_ID>
            <CLASSIFICATION_GROUP_NAME>Coaxial</CLASSIFICATION_GROUP_NAME>
            <CLASSIFICATION_GROUP_PARENT_ID>17010000</
CLASSIFICATION_GROUP_PARENT_ID>
        </CLASSIFICATION_GROUP>
    </CLASSIFICATION_GROUPS>
</CLASSIFICATION_SYSTEM>
```

Mapping Items to the Defined Classification Systems

You map items to the defined classification systems and feature groups using the `REFERENCE_FEATURE_SYSTEM_NAME` and the `REFERENCE_FEATURE_GROUP_ID` elements within the `ARTICLE_FEATURES` element.

Element	Mandatory	Type and length	Notes
`REFERENCE_FEATURE_SYSTEM_NAME`	No	String (50)	Specifies the classification system used to classify the article... Example:`<REFERENCE_FEATURE_SYSTEM_NAME>ETIM-1.0</REFERENCE_FEATURE_SYSTEM_NAME>`
`REFERENCE_FEATURE_GROUP_ID`	No	String (60)	Specifies the classification ID of the group associated with the specified classification system. Example: `<REFERENCE_FEATURE_GROUP_ID>17010100</REFERENCE_FEATURE_GROUP_ID>`

> **Note**
>
> When classifying items, it is mandatory that you describe the item using the default classification system defined for your solution or a classification system that is mapped to the default classification system.

The following example lists an item described according to two different classification systems.

```
<ARTICLE_FEATURES>
    <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</REFERENCE_FEATURE_SYSTEM_NAME>
    <REFERENCE_FEATURE_GROUP_ID>52152006</REFERENCE_FEATURE_GROUP_ID>
    <FEATURE>
        <FNAME>DIN Size</FNAME>
        <FVALUE>A4</FVALUE>
    </FEATURE>
    <FEATURE>
        <FNAME>Material</FNAME>
        <FVALUE>Plastic</FVALUE>
    </FEATURE>
    <FEATURE>
        <FNAME>Color</FNAME>
        <FVALUE>red</FVALUE>
    </FEATURE>
</ARTICLE_FEATURES>
<ARTICLE_FEATURES>
    <REFERENCE_FEATURE_SYSTEM_NAME>ETIM-1.0</REFERENCE_FEATURE_SYSTEM_NAME>
    <REFERENCE_FEATURE_GROUP_ID>17010100</REFERENCE_FEATURE_GROUP_ID>
    <FEATURE>
        <FNAME>Width</FNAME>
        <FVALUE>15</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
        <FEATURE>
        <FNAME>Height</FNAME>
        <FVALUE>8</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
    <FEATURE>
        <FNAME>Length</FNAME>
        <FVALUE>32</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
```

```
    <FEATURE>
        <FNAME>Color</FNAME>
        <FVALUE>red</FVALUE>
    </FEATURE>
</ARTICLE_FEATURES>
```

Non-Standard Customer Catalog Items

A non-standard item lacks attributes that the buying organization requires to identify it. Non-standard items are supported by Ariba Buyer 8.2 and later. They are not supported by SAP Ariba Buying solutions.

An example of a non-standard item is a customer catalog item without an explicit part number. Because a part number is a required field even for partial items, SAP Ariba Buying solutions assign a part number to the item. Part numbers are automatically included on requisitions and sent to suppliers on orders. In this case, use the `isInternalPartID` flag to indicate that the item is a non-standard item. `isInternalPartID` identifies these items in the supplier's system so that when processing orders, SAP Business Network knows to use the description and other attribute values to identify exactly the item ordered rather than relying on the part number.

Another example of a non-standard item is when a catalog contains partial items (for services or customizable items) that do not have supplier part numbers or they contain internal tracking numbers that differ from the supplier part numbers. The `isInternalPartID` flag indicates to SAP Business Network that the item is non-standard and lets SAP Business Network process the order despite any part number differences.

To use non-standard items in CIF catalogs, include the `isInternalPartID` field in the `FIELDNAMES` header of the catalog and then specify each non-standard item in the body of the catalog. In cXML, non-standard items are declared with the extrinsic tag. SAP Ariba Buying solutions do not currently support extrinsics in BMEcat catalogs.

Related Information

Additional Item Details in BMEcat Catalogs [page 59]
Related Items and Documents in Customer Catalogs [page 55]

Enabling Incremental Loading for Customer Catalog Items

When buying organizations receive CIF and cXML catalogs, Ariba Buyer and SAP Ariba Procurement solutions can either load entire files or load just the items that have changed. Loading only new or changed items is called incremental loading. Incremental loading is potentially faster than loading all items (full loading).

Additionally, different catalog formats provide different levels of incremental loading support:

Format	Support for Incremental Loading
CIF	• CIF offers full and incremental loading
cXML	• 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 and CIF, customer catalogs can use a `loadmode` attribute to indicate whether they are full or incremental. For CIF catalogs, use `F` (for Full) and `I` (for Incremental). For cXML catalogs, you must specify `Incremental` and `Full` (do not abbreviate).

If the `loadmode` attribute is missing from the customer catalog file, SAP Ariba solutions assume that it is full load and also generates a warning. To avoid these warnings, explicitly indicate `F` (for Full) on the `loadmode` attribute for full load customer catalogs.

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

Any item not marked for deletion is either an update or an addition. To determine which items are updates, procurement applications compare 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 `T` to mark an item for deletion. If the item is a regular add item, leave the `Delete` column blank for that item. For example, the following CIF file contains one delete item and one add (or update) item:

```
CIF_I_V3.0
LOADMODE:
     I

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,
     Delete

TIMESTAMP: 2005-03-29
UNUOM: TRUE
ITEMCOUNT: 2
DATA
58,A|2C-311F,C-311F,"NOTE,für POST-IT,1.5X2+DISP",33331111,2.85,EA,1,MMM-C,https://
i97.bcop.com/catalog_html/120703.htm,,2.84,
     T

    58,A;2C-311N,C-311N,"DISPENSER,F/310AN,PNOTES",33331111,2.85,EA,1,MMM-C,https://
i97.bcop.com/catalog_html/213300.htm,,2.84,en_US,
```

```
ENDOFDATA
```

In cXML, there is an explicit `IndexItemDelete` element that indicates an item is to be deleted. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.007/cXML.dtd">

     <Index loadmode="Incremental">

    <SupplierID domain="DUNS">942888711</SupplierID>
    <SupplierID domain="psoft">xYZ124</SupplierID>
    <IndexItem>

     <IndexItemDelete>

            <ItemID>
                <SupplierPartID>pn12356</SupplierPartID>
            </ItemID>

     </IndexItemDelete>

...
```

> **Note**
>
> If a customer catalog is specified as full load and contains delete items, the delete items generate errors.

Related Information

About CIF Catalogs [page 81]
About cXML Catalogs [page 105]

Searchable Fields for Customer Catalog Products and Services

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

Some fields are checked by default, and some fields need to be explicitly searched. You can provide data in the fields that are checked by default to improve the likelihood of your product or service being found.

Ariba Buyer and SAP Ariba Procurement solutions catalog searching is highly configurable. The default behavior is as follows.

When you search for a term, all these fields are searched:

- Supplier Name
- Supplier ID
- Supplier Part ID

- Manufacturer Name
- Manufacturer Part ID
- Description
- Short Name
- Keyword
- Parametric Data
- Parametric Name

Searchable from specific fields:

- Price
- ETA (Lead Time)
- Language

SAP Ariba Procurement solutions ignore all punctuation during searching. They remove accents from characters and any trailing “s” in the search term before searching.

SAP Ariba Procurement solutions administrators can create an equivalent words map, which enables users to perform synonym searching. Do not use use stop words when creating an equivalent words map. Stop words refer to words that are ignored during the search process. For example, if an English language user enters "`name plate`" while searching, incorrect results are displayed because the word "`name`" is a stop word. For information about equivalent words, see Equivalent Words. For information about stop words, see Stop Words.

Related Information

Enabling Incremental Loading for Customer Catalog Items [page 66]
Partial Items, Required Fields and IsPartial Field, and Enumeration Values [page 54]
Parametric Search Data [page 48]

Images for CustomerCatalog Items

Ariba Buyer 8.2.2 or later and SAP Ariba Procurement solutions can display images for customer catalog items. You can add customer catalog images to give requisitioners a visual representation of your products. They provide customers with a rich catalog-browsing experience, similar to buying products on the Web.

Methods for Linking to Images in Customer Catalogs [page 69]
Thumbnail, Detail, and Full-Size Images in CIF and cXML Catalogs [page 71]
Thumbnail, Detail, and Full-Size Images in BMEcat Catalogs [page 71]
Multiple Images in CIF Catalogs [page 72]
Format, Size, and Composition for Customer Catalog Images [page 74]
Website Security for Customer Catalog Images [page 75]
Thumbnail, Detail, and Full-Size Images in Excel Catalogs [page 75]

Methods for Linking to Images in Customer Catalogs

Images don't reside in your customer catalogs. Instead, each image resides in a separate file. You insert a link in your customer catalogs to each image.

There are two link methods you can use:

- Local Images—for example, `lawnmower455.jpg`. Customer catalog administrators manually load image files into their systems when they load customer catalogs. Send your image files to your customers so they can store them locally. You cannot distribute them through SAP Business Network; send them directly to your customers. Most suppliers send them by using FTP or external file transfer.
  Provide relative pathnames in your customer catalogs; the image files you provide must have the same directory structure. For example, if your customer catalog contains "`/catalog/images/lawnmower455.jpg`" the images you provide must reside in a /catalog/images directory.
  You can send your image files in zip archives. Procurement applications can automatically unzip image files when they import images.
  If you choose to zip the folder containing the images, then the paths of the image files in the customer catalog file must contain the folder name. For example, if the catalog image (`test_image.jpg`) and the thumbnail image (`test_image_sm.jpg`) files are located in a folder named TestImages and you zip the folder then the paths of the image files in the CIF file must be specified as follows:

```
sid502,S100093a,,"image-test, Pack
of 10, Rev.",,0.01,PK,3,,http://cp.suntrust.com/poEforms/
100093.PDF,,,USD,,,en_US,,,,,{UNSPSC_V12.2=141118;},,Equity Funds A C I
Shares,,,false,
       TestImages/test_image.jpg,TestImages/test_image_sm.jpg
      ,,,,,,
```

  The images aren't displayed if you do not mention the folder name in the path.
  The following CIF example shows two items with full-size and thumbnail local file name images.

```
CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC_V13.5
CURRENCY: USD
SUPPLIERID_DOMAIN: NetworkID
TIMESTAMP: 2006-07-08 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,
       Image, Thumbnail

DATA
AN1000007581,S066,S066,"Loveseat, Upholstered in Grade 3 Fabric, Natural
Color",561017,540.00,EA,6,"Arcon Upholstery",,,0,
       "008.jpg","008THUMB.jpg"

AN1000007581,S036,S036,"Club Chair, Upholstered in Grade 3 Fabric, Natural
Color",561017,395.00,EA,6,"Arcon Upholstery",,,0,
       "aero.jpg","aeroTHUMB.jpg

ENDOFDATA
```

> **Note**
>
> If your customer (buying organization) has content management system (CMS) configured, you can import local images using the **Images and Attachments File** option while uploading the customer catalog to SAP Business Network.

- URL Images—for example, `https://www.lawnsRUs.com/images/lawnmower455.jpg`. If you provide URLs, you must store your images on a publicly-accessible website so that procurement applications can retrieve them and cache them locally. Your image website must be available all the time. The URL can contain a query or an automated script (for example: `https://www.lawnsRUs.com/scripts/search.asp?find=lawnmower.jpg`), as long as a single image is retrieved, not other information such as text or HTML.
  Procurement applications periodically refresh images by fetching them again. By default, SAP Ariba solutions refresh images after 30 days.
  Before using URLs, verify that your customers have enabled their procurement systems to access remote websites.

The following cXML sample shows a customer catalog item with a full-size image and a thumbnail image, both referenced as URLs:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.014/cXML.dtd">
<Index>
    <SupplierID domain="NetworkID">AN1000007581</SupplierID>
    <IndexItem>
        <IndexItemAdd>
            <ItemID>
                <SupplierPartID>S066</SupplierPartID>
            </ItemID>
            <ItemDetail>
                <UnitPrice>
                    <Money currency="USD">540.00</Money>
                </UnitPrice>
                <Description xml:lang="en">
                    Loveseat, Upholstered in Grade 3 Fabric, Natural Color
                </Description>
                <UnitOfMeasure>EA</UnitOfMeasure>
                <Classification domain="UNSPSC_V13.5">561017</Classification>

                     <Extrinsic name="Image">


                         https://www.arcon.com/catalog/images/008.jpg


                     </Extrinsic>


                     <Extrinsic name="Thumbnail">


                         https://www.arcon.com/catalog/images/008THUMB.jpg


                     </Extrinsic>

            </ItemDetail>
            <IndexItemDetail>
                <LeadTime>6</LeadTime>
            </IndexItemDetail>
        </IndexItemAdd>
    </IndexItem>
</Index>
```

Related Information

Images for CustomerCatalog Items [page 68]
Format, Size, and Composition for Customer Catalog Images [page 74]
Multiple Images in CIF Catalogs [page 72]

Thumbnail, Detail, and Full-Size Images in CIF and cXML Catalogs

You can provide thumbnail images, full-size images, or both types of images. Procurement systems display thumbnails immediately after users search and full-size images if users click the thumbnails. Thumbnail and full-size images do not have to show the same image.

If users click that item, they see the following figure containing the full-size image. Clicking **Enlarge** displays your original image in its own window.

Use the catalog field names "`Image`" for full-size images and "`Thumbnail`" for thumbnails. In cXML catalogs, use these names as the name of `Extrinsic` elements.

If you provide only full-size images, procurement systems scale them to create thumbnail images. However, if you provide only thumbnail images, procurement systems cannot scale them to full-size images. In this case, they use the thumbnail image in the scale you provide after users search and when users request more information.

Thumbnail, Detail, and Full-Size Images in BMEcat Catalogs

You can provide thumbnail, detail, and full-size images for BMEcat catalog items. The SAP Ariba solution can display multiple thumbnail and detail images alongside the full-size main image; users can click the thumbnail and detail images to enlarge them.

`MIME_INFO` for a BMEcat catalog item can contain multiple `MIME` fields with images. SAP Ariba solutions support the following mime types for images:

- `image/jpeg`
- `image/gif`

SAP Ariba solutions support the following mime purposes for images:

- `thumbnail`
- `normal`
- `detail`

Images can be stored locally, or they can be hosted externally at a URL. Although there is no limit to the number of images an item can have, SAP Ariba recommends no more than 10 images per item for the best display. The SAP Ariba solution resizes normal or detail images to thumbnails if their defined purpose is `thumbnail`. If the same image is used for multiple image types, they should all have the same `MIME_ORDER`.

When users click on an image to view an enlarged version, the detail image is displayed.

> **Note**
>
> If you specify a thumbnail image for an item, you must also specify an associated normal or detail image. SAP Ariba solutions do not allow catalog items that only have a thumbnail image.

The following example defines a full-size, detail, and thumbnail image for an item:

```
<MIME_INFO>
    <MIME>
        <MIME_TYPE>image/jpeg<MIME_TYPE>
        <MIME_SOURCE>http://12.34.56.789:8888/images/123lawnmower_full.jpeg</
MIME_SOURCE>
        <MIME_DESCR>Lawnmower 123 main view</MIME_DESCR>
        <MIME_ALT>Lawnmower 123 main view</MIME_ALT>
        <MIME_PURPOSE>normal</MIME_PURPOSE>
        <MIME_ORDER>1</MIME_ORDER>
    </MIME>
    <MIME>
        <MIME_TYPE>image/jpeg</MIME_TYPE>
        <MIME_SOURCE>http://12.34.56.789:8888/images/123lawnmower_detail.jpeg</
MIME_SOURCE>
        <MIME_DESCR>Lawnmower 123 engine details</MIME_DESCR>
        <MIME_ALT>Lawnmower 123 engine details</MIME_ALT>
        <MIME_PURPOSE>detail</MIME_PURPOSE>
        <MIME_ORDER>2</MIME_ORDER>
    </MIME>
    <MIME>
        <MIME_TYPE>image/jpeg</MIME_TYPE>
        <MIME_SOURCE>http://12.34.56/789:8888/images/
123lawnmower_side.jpeg<MIME_SOURCE>
        <MIME_DESCR>Lawnmower 123 side view</MIME_DESCR>
        <MIME_ALT>Lawnmower 123 side view</MIME_ALT>
        <MIME_PURPOSE>thumbnail</MIME_PURPOSE>
        <MIME_ORDER>3</MIME_ORDER>
</MIME_INFO>
```

> **Note**
>
> Although the BMEcat DTD does not require `MIME_PURPOSE`, the SAP Ariba solution uses it to correctly identify and display images. The SAP Ariba solution ignores all items that do not have a `MIME_PURPOSE` for each image during customer catalog import.
>
> The SAP Ariba solution enforces a character limit of 250 each for `MIME_SOURCE` and `MIME_DESCR` and of 50 for `MIME_ALT`.

Multiple Images in CIF Catalogs

Items can have multiple images associated with them, including thumbnails or detail images; those images can be stored locally or hosted on external servers and referenced via URL.

SAP Ariba solutions can display these multiple thumbnail and detail images alongside the full-size main image; users can click the thumbnail and detail images to enlarge them.

You can provide multiple thumbnail images using the `AuxiliaryThumbnails` field and the detail, and full-size images using the `AuxiliaryImages` field. You can specify the multiple images as a hash table of [“key-value”] pairs. The syntax is `[{"key"="value";"key"="value";"key"="value" * };{ * }]`. The string within braces ({}) indicates one image.

For example, `[{"mimetype"="image/jpeg";"mimesource"="http://10.138.169.202:8888/myapp/main.jpeg";"mimedescr"="CM 44 Xperia Z2";"mimealt"="Front view";"mimepurpose"="thumbnail";"mimeorder"="1"}]`

The following is an example of an `AuxiliaryThumbnails` field value:

```
[{"mimetype"="image/jpeg";"mimesource"="http://10.138.169.202:8888/myapp/
main.jpeg";"mimedescr"="CM 44 Xperia Z2";"mimealt"="Front
view";"mimepurpose"="thumbnail";"mimeorder"="1"}]
```

The following is an example of an `AuxiliaryImages` field value:

```
[{"mimetype"="image/jpeg";"mimesource"="http://10.138.169.202:8888/myapp/
moto_aux1.jpeg";"mimedescr"="CM 44 Xperia
Z2";"mimealt"="Front view";"mimepurpose"="normal";"mimeorder"="1"};
{"mimetype"="image/jpeg";"mimesource"="http://10.138.169.202:8888/myapp/
moto_aux1.jpeg";"mimedescr"="CM 44 Xperia
Z2";"mimealt"="Front view";"mimepurpose"="detail";"mimeorder"="1"};
{"mimetype"="image/jpeg";"mimesource"="http://10.138.169.202:8888/myapp/
moto_aux2.jpeg";"mimedescr"="Black color";"mimealt"="Front
view Black color";"mimepurpose"="normal";"mimeorder"="2"};
{"mimetype"="image/jpeg";"mimesource"="http://10.138.169.202:8888/myapp/
moto_aux2.jpeg";"mimedescr"="Black color";"mimealt"="front
view black color";"mimepurpose"="detail";"mimeorder"="2"};
{"mimetype"="image/jpeg";"mimesource"="http://10.138.169.202:8888/myapp/
moto_aux3.jpeg";"mimedescr"="Display";"mimealt"="Display
screen";"mimepurpose"="normal";"mimeorder"="3"};{"mimetype"="image/
jpeg";"mimesource"="http://10.138.169.202:8888/myapp/
moto_aux3.jpeg";"mimedescr"="Display";"mimealt"="Display
screen";"mimepurpose"="detail";"mimeorder"="3"};]
```

The following table lists the various keys and their supported values to define multiple images:

Key Name	Required/Optional	Notes
`mimetype`	Optional	Indicates the type of the image file. SAP Ariba solutions support the following values for thumbnail, detail, and full-size images: • `image/jpeg` • `image/gif`
`mimesource`	Required	Indicates the source of the image. The value can be a URL or the pathname to an image.
`mimedescr`	Optional	Indicates the description of the image.
`mimealt`	Optional	Indicates the alternative text for the image.

Key Name	Required/Optional	Notes
`mimepurpose`	Optional	Indicates the purpose of the image. SAP Ariba solutions support the following values: • `thumbnail` for thumbnail images • `normal` for standard-size images • `detail` for full-size images **Note** If you specify a thumbnail image for an item, you must also specify an associated normal or detail image. SAP Ariba solutions do not allow customer catalog items that only have a thumbnail image.
`mimeorder`	Optional	Indicates the order in which the images are to be displayed. If the same image is used for multiple image types, they should all have the same `mimeorder` value.

Format, Size, and Composition for Customer Catalog Images

Images must be in JPEG, GIF, PNG, BMP, or TIFF format (for CIF or cXML catalogs) or JPEG or GIF format (for BMEcat catalogs). They can have any filename, but they must have the appropriate filename extension. Most suppliers use the part number as the filename. The file names and filename extension for images are case-sensitive.

You can zip images files into zip archives and send them to your customers. Zip archives must have a `.zip` filename extension. Image and zip filenames must contain only ASCII characters (without accents). Do not use compound images, which are several images loaded into one file.

Table 1: Recommended Sizes for Images

Image Type	Size (Pixels)
Full Size	250 x 250
Thumbnail	64 x 64
Image Size	90 KB

SAP Ariba solutions do not resize the original images that you provide in CIF or cXML catalogs. It does for BMEcat catalog images. The recommended file size is less than 50 KB.

Your images can be in black-and-white or in color. They can be photographs or drawings. Do not use animated images. For best visual results, images should have a neutral background.

Note

The maximum upload zip file size limit for customer catalog images is 300 MB.

Related Information

Images for CustomerCatalog Items [page 68]

Website Security for Customer Catalog Images

If your images reside on a website, you can make them more secure by requiring all connections to connect using HTTPS.

HTTPS is a secure form of the HTTP protocol. To enable HTTPS connections, install a web server digital certificate signed by a certificate authority trusted by your customer's procurement application. Then configure your web server to reject all regular http connections.

Related Information

Images for CustomerCatalog Items [page 68]

Thumbnail, Detail, and Full-Size Images in Excel Catalogs

You can provide thumbnail, detail, and full-size images for Excel catalog items. The SAP Ariba solution can display multiple thumbnail and detail images alongside the full-size main image; users can click the thumbnail and detail images to enlarge them.

Images can be stored locally, or they can be hosted externally at a URL.You define thumbnail, detailed, and full-sie images for items in Excel catalogs using the `Image` key and the `Thumbnail`, `Detailed`, and `Normal` attributes. If you specify an image then all the three attribute values must be provided.

You can specify additional images for an item by adding additional columns and naming them Image-1, Image-2, Image-3, and so on. A maximum of 10 images per item can be specified.

Item Masters in Customer Catalog

Item masters are customer catalog items from buying organizations that are not associated with a supplier and supplier part number. The buyer part number and facility combination that make up an item master is called a buyer item master.

Buyer item masters can be thought of as the representation of item masters inside of SAP Ariba solutions. Each buyer item master is associated with one or more supplier part numbers, creating cross-reference links. When a buyer item master containing active cross-references is created and saved, it is immediately available to users in the SAP Ariba solution's product catalog.

Cross-references to buyer item masters can be created automatically during the customer catalog import process by adding them to customer catalog files before importing them through the Catalog Manager in Ariba Administrator. Cross-reference links are defined in customer catalog files through the `LinkedBuyerPart` and `LinkedBuyerFacility` supplemental attributes. For example:

```
...
<TypeAttribute name="LinkedBuyerPart" type="Id"
 <Name xml:lang="en-US">Linked Buyer Part</Name>
</TypeAttribute>
<TypeAttribute name="LinkedBuyerFacility" type="Id">
 <Name xml:lang="en-US">Linked Buyer Facility</Name>
</TypeAttribute>
```

Then two additional columns referencing the attributes are added the customer catalog files. For more information about supplemental attributes, see Type Definition Files [page 156]

For more information on the item master file, see the Procurement data import and administration guide.

Related Information

CIF Catalogs [page 81]

Hazardous Material Labels

You can configure guided buying to display hazardous material information for applicable catalog items, which allows users to identify which items contain hazardous materials and the types of hazardous materials present.

This feature displays the hazardous materials present in items according to the Globally Harmonized System (GHS) of Classification and Labeling of Chemicals. You or your suppliers can add GHS codes in CIF, cXML, Excel, or BMEcat catalogs.

Use the following GHS codes in catalogs to display the hazardous material symbols on items.

GHS Code	GHS Code Name	GHS Code Symbol
GHS01	Explosives	

GHS Code	GHS Code Name	GHS Code Symbol
GHS02	Flammable	
GHS03	Inflammatory	
GHS04	Gases under pressure	
GHS05	Corrosive	
GHS06	Acute Toxicity	
GHS07	Irritant	

GHS Code	GHS Code Name	GHS Code Symbol
GHS08	Health Hazard	
GHS09	Environment Hazard	

You configure hazardous material details before you load your suppliers' catalogs in your SAP Ariba Buying solution. To configure this feature, do the following:

- 1. Work with your SAP Ariba Catalog Expert to extend the type definition files that contains the **HazMatClasification** attribute with the GHS codes (**GHS01**, **GHS02**, etc.). The following type definition file displays the new extensible type **HazMatClassification**:

```
<TypeDefinition>
<TypeProvider name="System">
<OrganizationID>
<Credential domain="NETWORKID">
<Identity>2</Identity>
</Credential>
</OrganizationID>
</TypeProvider>
<Comments/>
<Type name="CatalogItem" extends="System:CatalogItemInternal">
<Name xml:lang="en-US">Catalog Item</Name>
<Name xml:lang="fr">Article de Catalogue</Name>
<Name xml:lang="de">Katalogeinzelteil</Name>
<Description xml:lang="en-US">Type describing Supplemental attributes of any
catalog item data</Description>
<TypeAttribute name="HazMatClassification" type="Name" isRefinable="yes"
isSearchable="yes" isCollection="yes">
<Name xml:lang="en_US">HazMatClassification</Name>
<EnumerationValue>
<Name xml:lang="en_US">GHS01</Name>
<Extrinsic name="CanonicalValue">GHS01</Extrinsic>
</EnumerationValue>
<EnumerationValue>
<Name xml:lang="en_US">GHS02</Name>
<Extrinsic name="CanonicalValue">GHS02</Extrinsic>
</EnumerationValue>
<EnumerationValue>
<Name xml:lang="en_US">GHS03</Name>
<Extrinsic name="CanonicalValue">GHS03</Extrinsic>
</EnumerationValue>
<EnumerationValue>
<Name xml:lang="en_US">GHS04</Name>
<Extrinsic name="CanonicalValue">GHS04</Extrinsic>
</EnumerationValue>
```

```
<EnumerationValue>
<Name xml:lang="en_US">GHS05</Name>
<Extrinsic name="CanonicalValue">GHS05</Extrinsic>
</EnumerationValue>
<EnumerationValue>
<Name xml:lang="en_US">GHS06</Name>
<Extrinsic name="CanonicalValue">GHS06</Extrinsic>
</EnumerationValue>
<EnumerationValue>
<Name xml:lang="en_US">GHS07</Name>
<Extrinsic name="CanonicalValue">GHS07</Extrinsic>
</EnumerationValue>
<EnumerationValue>
<Name xml:lang="en_US">GHS08</Name>
<Extrinsic name="CanonicalValue">GHS08</Extrinsic>
</EnumerationValue>
<EnumerationValue>
<Name xml:lang="en_US">GHS09</Name>
<Extrinsic name="CanonicalValue">GHS09</Extrinsic>
</EnumerationValue>
<Extrinsic name="inSearchView">yes</Extrinsic>
<Extrinsic name="inDetailView">yes</Extrinsic>
</TypeAttribute>
<TypeAttribute name="CnC" type="Flag">
<Name xml:lang="en_US">Click and Collect</Name>
<Extrinsic name="inDetailView">yes</Extrinsic>
</TypeAttribute>
</Type>
</TypeDefinition>
```

2. Add the GHS codes to your catalogs and then upload them to your SAP Ariba Buying solution. Guided buying interprets the codes and displays the applicable GHS symbol.

> **Note**
>
> Your suppliers can also add the GHS codes to your catalogs.

The following catalog examples include items with the hazardous material attributes:
Example: CIF catalog with hazardous material attributes:

```
CIF_I_V3.0
LOADMODE: F
SUPPLIERID_DOMAIN: buyersystemid
COMMENTS: Generated on Thu Mar 20, 2025
UNUOM: TRUE
CURRENCY: USD
CHARSET: UTF-8
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,
PunchOut Enabled, Supplier Part Auxiliary ID, Language, Parametric
Name, Parametric Data, Expiration Date, Effective Date, Classification
Codes, Territory Available, Short Name, IsPartial, IsInternalPartId,
Delete, Image, Thumbnail, InKitOnly, PunchoutLevel, DefaultRelevance,
Price Unit Quantity, Price Unit, Unit Conversion, Price Unit Quantity
Description, Promotion Rank, Is Preferred Item, storagelocations, plant, cnc,
hazmatclassification, stockidentifier, materialnumber, materialgroup, green,
bestprice, hazardousmaterials
DATA
"sid496","Part123","Part123","Monitor Adapter",,"222.22","EA","13","Hallmark
Cables","s.html","config/SampleImages/
hallmark_cables.html","34","USD","false",,"en_US",,,,,"{UNSPSC=43211800;}",,,"
false","false","false",,,"false",,,,,,,,"false",,,"false","(ghs01,ghs02,ghs04)
",,,,"false","false","true"
ENDOFDATA
```

→ Remember

Follow these guidelines when implementing this attribute type:

- Use only the GHS code in your catalog file, not the GHS name.
- Use only English for the **`HazMatClassification`** type attribute.
- The symbols used to identify each type of hazard cannot be changed as per European Union (EU) laws.

For more information on configuring your catalog, see Customer Catalogs in Guided Buying.

CIF Catalogs

Basics of CIF Catalog Files [page 81]

Supplier IDs in Customer Catalogs [page 96]

CIF Catalog Examples [page 99]

CIF File Editing [page 102]

Basics of CIF Catalog Files

Catalog Interchange Format (CIF) is a general-purpose customer catalog description format. It is used by thousands of procurement systems from many vendors around the world.

CIF catalog files list items, their descriptions, and prices. Every CIF file has three sections: header, body, and trailer.

The header specifies general attributes that apply to all the data in the file.

The body describes the items in the catalog.

The trailer is the last line in the file and marks the end of the catalog.

There are variations based on the CIF version you implement, and some of the settings in the header can be overwritten at the individual line level within the body.

The following example is a customer catalog file containing a note dispenser with the supplement attribute fields needed to create a cross-reference link:

```
CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
COMMENTS: This is a sample CIF 3.0 file for loading into the catalog server
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,
     LinkedBuyerPart
    ,
     LinkedBuyerFacility

TIMESTAMP: 2000-03-29
UNUOM: TRUE
ITEMCOUNT: 1
DATA
58,A2C-311N,C-311N,"DISPENSER,F/310AN,PNOTES",44111501,2.85,EA,1,MMM-C,https://
i97.bcop.com/catalog_html/213300.htm,,2.84,"
     bim100
    ","
     ATL
    "
ENDOFDATA
```

If the buyer item master is already created in Ariba Administrator, when this customer catalog file is activated, the link to the note dispenser appears automatically in the buyer item master. For more information about item masters and buyer item masters, see the *Procurement Data Import and Administration Guide*.

If you import multi-supplier CIF catalogs, you must use image URLs and not a compressed file with images. Also, for a multi-supplier CIF catalog, when a user clicks the **Refresh Images** button, only images from the supplier selected while uploading the catalog are refreshed and the images from the other suppliers aren't refreshed.

Similar customer catalog items can refer to the same catalog image. If the customer catalog link is not valid, the procurement application displays **Image Not Available**.

Catalog Filenames

Filenames for customer catalogs can contain any standard ASCII characters. You must use the extension `.cif` for CIF catalogs.

SAP Ariba recommends that you use the same filename for all versions of a given catalog to make customer catalog management easier for buying organizations.

Excel-to-CIF Converter

SAP Business Network has a built-in Excel to CIF file converter that allows you to create and maintain data in an Excel spreadsheet and easily convert it to CIF format.

For more information, see *Creating and managing customer catalogs*.

CIF Header Syntax and Keys

The CIF header defines attributes that apply to the entire file. SAP Business Network recognizes a standard set of header keys.

CIF Header Syntax

The following example is a valid, fully qualified CIF file 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: 2005-04-18 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, IsPartial
```

The first line of the file always declares the CIF version. It must appear at the beginning of every CIF file.

Every line after the header is a key that employs a standard name: value pair, where name is the name of the key, and value is the unique data for the key in the CIF file:

```
KEY:value
```

Each new line designates a separate header key. Keys can appear in any order. All white space is significant. Do not include spaces, unless they are within quotes (`""`).

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

> **Note**
>
> Although keys are case-insensitive, they are typically written in caps for legibility and to aid in file maintenance. SAP Ariba recommends that you follow this convention in your files.

CIF Header Keys

The following table lists the keys recognized in the CIF catalog file header. Other keys are permitted, but are silently ignored by SAP Business Network.

Key Name	Type	Description
`CIF_I_V3.0`	Required	Default: *n/a* Specifies the CIF version, specify one.
`CURRENCY`	Required	Default: **USD** Specifies the currency for interpreting prices in the current file. Must be an ISO 4217 currency name. Individual items can override this value by using the `Currency` body field. Example: **`CURRENCY:USD`**
`CHARSET`	Required	Default: `Cp1252` (Windows Latin-1) Specifies the ISO character set used in this file. Example: **`CHARSET:UTF-8`**

Key Name	Type	Description
CODEFORMAT	Required (Optional when using xlsx)	Default: UNSPSC Specifies the default commodity code system. The SAP Ariba solution recognizes an optional version number. You can supplement individual items with additional codes by using the `Classification Codes` field in the body. Example: `CODEFORMAT:UNSPSC` `CODEFORMAT:UNSPSC_V7.1`
COMMENTS	Optional	Default: n/a Indicates a line of plain text. **Note** Liberal use of comments can aid in later maintenance efforts. A good comment explains why you are doing something, not just what you are doing. Do not include XML tags in a `COMMENTS` value. Doing so results in parse errors. Example: `COMMENTS: This catalog is for specialized number 2.5 pencils with red paint.`
DUNS	Optional (Required if backward compatibility is needed)	Default: n/a Supported values: TRUE or FALSE Indicates whether supplier IDs are D-U-N-S numbers. Current versions of SAP Ariba solutions do not recognize this attribute. It is available for backward compatibility. New catalogs must use the `SUPPLIERID_DOMAIN` attribute instead. Example:DUNS:TRUE
UNUOM	Required	Default: FALSE Supported values: TRUE or FALSE Indicates whether this file uses United Nations Units of Measure (UNUOM). If unspecified or set to FALSE, the file is assumed to use American National Standards Institute (ANSI) units of measure. Example: `UNUOM:TRUE`
SUPPLIERID_DOMAIN	Required	Default: n/a Specifies the domain for supplier IDs used in this file. Typical domains are DUNS and `NETWORKID`. Example: `SUPPLIERID_DOMAIN:NETWORKID`

Key Name	Type	Description
`FIELDNAMES`	Required	Default: none Declares the fields in the body of the file. Fields beyond the first 12 are optional, and can be declared in any order. You can declare your own custom fields, but procurement applications ignore them. `FIELDNAMES` can only contain ASCII characters. The names must match the names of the available fields exactly, including capitalization. If you do not include `FIELDNAMES`, the body must contain only the first 12 fields. Example: `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, Language, Parametric Data, Parametric Name`
`LOADMODE`	Required	Default: `F` Supported values: `F` or `I` Declares whether the catalog is complete (Full) or a partial (Incremental) update. Example: `LOADMODE:F`
`TIMESTAMP`	Required	Default: none `YYYY-MM-DD hh:mm:ss`, where the time is in 24-hour format.
`ITEMCOUNT`	Optional	Default: none Number of items in this file. Used for error checking.

CIF Body Syntax

The body of a CIF file is a series of entries in a Comma Separated Value (CSV) format. The body is delimited by the keywords `DATA` and `ENDOFDATA`. Each of these keywords must be on its own line.

If you use the wizard on the customer catalog dashboard, the customer catalog file cannot exceed 95 MB or 400,000 lines. If you use the cXML `CatalogUploadRequest` transaction, it cannot exceed 10 MB.

When publishing CIF catalogs in an Excel (.xls) format on SAP Business Network ensure that the file does not exceed 1 MB.

Basic CIF Body Syntax Rules

Basic CIF body syntax rules are simple and structured.

1. Each new line is a separate entry.
   All white space is significant. Do not include spaces, unless they are within quotes (`""`). Blank lines are not allowed. Line returns always indicate the end of a line and are not allowed elsewhere within the line, even inside quoted text.
2. Commas are significant.
   Each comma (`,`) means "go to the next field." You can leave fields empty by inserting two commas immediately next to each other with no space in between. For example:

   ```
   Field1,Field2,,Field4
   ```

   If you have a text field that contains a comma, you must insert quotation marks around the field values. For example, if you wanted to include the following text in a field:

   ```
   Field1,hammer, wood,Field3
   ```

   You must surround the whole field in quotation marks:

   ```
   Field1,"hammer, wood",Field3
   ```

   Commas used as decimal separators in numbers require you to quote the entire number:
   `Field1,"7,25",Field3`
3. Quotation mares within quoted fields must be escaped.
   To use a quotation mark(`"`) within a quoted field, use two quotation marks. If the field is not quoted, do not escape the quotation mark. Both of the following examples are legal:

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

   Single quotation marks (') need no special treatment.
4. Decimal and currency values may be delimited using either commas or periods.
   You can use either a period (`.`) or comma (`,`) as a decimal separator. Spaces are not permitted. To use a comma as a decimal separator, quote the entire number. For example, to specify a price of three dollars, you can write either `3.00` or `"3,00"`, but not `3 00`.
   Do not use a comma, period, or space as a thousands separator. For example, to write three thousand dollars, write `3000.00` or `"3000,00"` but not `"3,000.00"`
5. Currency symbols must be omitted when specifying currency values.
   Do not include currency symbols, such as $, ¥, or £.
   The procurement application determines the correct localized display format, for example, `$4.20` or `¥850.`

CIF Body Row Field Rules

Each row in the body must have the same number of fields even if that means specifying that a field is empty. To specify an empty field, enter nothing in the field except the separating commas.

CIF has 12 fields in a fixed order and additional optional fields, which can appear in any order after the fixed set of fields. You name the optional fields using the `FIELDNAMES` key in the header and must name every new column.

You can add your own custom fields (supplemental attributes) to CIF catalogs. SAP Business Network accepts CIF catalogs that have any number of fields. Most procurement applications only recognize the standard CIF fields, and discard unrecognized fields during catalog import.

> **Note**
>
> An SAP Ariba solution can be configured to recognize custom fields as supplemental attributes. For more information, see Supplemental Attributes [page 47].

CIF Trailer (End of File) Rules

The last line in a CIF file is `ENDOFDATA`. Applications look for this keyword to determine if the ned of the file has been reached. Applications ignore any data after the `ENDOFDATA` line.

Required and Optional CIF Body Row Fields

Each row in the body of all CIF files must contain 12 required fields in precise order. Some of these required fields may be left blank (delimited by commas), but must be included. Additional optional fields can be appended to the end of each row after the required fields.

Required CIF Body Row Fields

The first 12 body row fields are required and must appear in order. Some of these fields can be empty, but empty fields must be delimited by commas.

The following table lists the required fields in order, describes each field, indicates if a field value is required or can be left blank, the field's data type and the maximum field length.

Order	Field Name	Field Value Required/ Optional	Data Type and Maximum Length	Description
1	`Supplier ID`	Required	String 255	Default: n/a. Typically a D-U-N-S or NetworkID number. SAP Ariba solutions support up to 128 characters and accepts multi-supplier CIF catalogs. This string is case-insensitive. Example:`942888710`

Order	Field Name	Field Value Required/ Optional	Data Type and Maximum Length	Description
2	`Supplier Part ID`	Required	String 255	Default: n/a. Part identifier.SAP Ariba solutions support up to 128 characters. This string is case-insensitive. Never use the characters ?, {, or } in Supplier Part IDs, as they cause validation errors. Example:`SUX-600`
3	`Manufacturer Part ID`	Optional (must be comma delimited)	String 255	Can be empty. Part identifier used by manufacturer. SAP Ariba solutions support up to 128 characters. This string is case-insensitive. Example:`286-33`
4	`Item Description`	Required	String 1000	Default: n/a. Description of the product or service. SAP Ariba solutions support up to 2000 bytes; if you are using Unicode strings, this means 1000 characters. Example: `"Super Duper 33-MHz 286, 4MB RAM, 10MB HD, 14"" VGA Color Monitor"`
5	`SPSC Code`	Required	String 40	Default: n/a. Product commodity 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. **Note** USPSC codes typically have a period between every two digits: `44.12.21.04`; catalog files require these codes to contain no punctuation, for example, `44122104`. Each item must have at least one code, either here or in the `Classification Codes` field. SAP Ariba solutions accept up to 128 characters. This string is case-insensitive. Example:`43171803`

Order	Field Name	Field Value Required/ Optional	Data Type and Maximum Length	Description
6	`Unit Price`	Required	Decimal	Default: n/a. Customer's price. See Decimal and currency values may be delimited using either commas or periods. [page 86] for details. No value required for PunchOut or Partial items. This string is case-insensitive. Examples: `1259.59 "17,00"` **Note** Ariba Network requires a price in either the Unit Price or Market Price field. If the unit price is not specified, the available market price is used.
7	`Unit of Measure`	Required	String 32	Default: n/a. UN or ANSI X.12 standard unit of measure. No value required for PunchOut or Partial items. This string is case-insensitive. Example: EA
8	`Lead Time`	Optional (must be comma delimited)	Integer	Can be empty. Number of business days between the time the order is received and the time it is delivered to the customer. Example: 3
9	`Manufacturer Name`	Optional (must be comma delimited)	String 255	Can be empty. Name of the product's manufacturer. Example: `"Super Computer"`
10	`Supplier URL`	Optional	String 255	Can be empty. Web address of the supplier's data sheet. Example: `http://www.slowcomp.com/products/indepth.htm`
11	`Manufacturer URL`	Optional	String 255	Can be empty. Web address of the manufacturer's data sheet. Example: `http://www.slowcomp.com/products/dataindepth.htm`
12	`Market Price`	Optional	Decimal	Can be empty. List price or suggested retail price. This field has the same requirements as `Unit Price` and is used for comparison only. It is not displayed in the SAP Ariba solution's customer catalog user interface. Example: `1599.49`

Optional CIF Body Row Fields

CIF supports optional body row fields. When you choose to use any of these fields, you must declare the field names as values in the `FIELDNAMES` key in the CIF header.

The order you specify optional field names in the FIELDNAME key determines the order in which you must enter the field data in your body rows.

The following table lists optional CIF 3.0 fields.

Field Name	Type and Length	Description	CMS Required?
`Short Name`	String 80 (SAP Business Network supports up to 50 characters)	Short title for the item, to be used in lists where `Item Description` is too long. If not specified, the first 80 characters of the `Item Description` will be displayed in the user interface. Example: `Laptop Computer`	No
`Language`	String 255 (SAP Ariba solutions support up to 128 characters)	Java locale code for the language used in the `Item Description`. Use an underscore (_) instead of a dash (-) in a CIF catalog. This string is case insensitive. If you have multiple items in different languages, differentiate them by using the locale code as the `Supplier Part Auxiliary ID`. For the list of Java locale codes, see: http://java.sun.com/j2se/1.4.2/docs/guide/intl/locale.doc.html **Note** Do not use ISO 3166-1 and 3166-2 territory codes here. Example: `fr_FR`	No
`Currency`	String 32	ISO 4217 currency code. This value overrides `Currency` in the header. **Note** Do not use ISO 3166-1 and 3166-2 codes here, those codes are territory codes. SAP Ariba solution support up to five characters. This string is case-insensitive. Example: **USD**	No
`Expiration Date`	Date	Date that the item is no longer valid, in `YYYY-MM-DD` format. This date must be in the future. Example: `2005-12-01`	No
`Effective Date`	Date	Date that the item becomes valid, in `YYYY-MM-DD` format. This date must be prior to the expiration date. Example: `2005-01-25`	No
`Classification Codes`	String 64	Hash table of ["domain-value"] pairs. The procurement application uses the first key that it recognizes. This string is case-insensitive. Example: `{UNSPC=8BE100}`	No

Field Name	Type and Length	Description	CMS Required?
`Parametric Data`	String 255	Hash table of ["key-value"] pairs. Entries (parametric attribute names) can be up to 128 characters, and cannot be null. This string is case-insensitive. To learn more, see Parametric Search Data [page 48]. Example: `{TYPE=SEAMLESS;SIZE=20;"WEIGHT PER FOOT"=94.00;}`	No
`Parametric Name`	String Ariba Buyer 7.0: 24 Ariba Buyer 7.1 and later: 128	Textual label for the type of data you supplied in the `Parametric Data` field. Required if you specify `Parametric Data`. This string is case-insensitive. Example: `PIPES`	No
`PunchOut Enabled`	Boolean	Specifies a PunchOut index catalog item. The default value is `False`. This value is case-insensitive. Example: `True`	No
`PunchoutLevel`	String	Determines how the procurement application displays the PunchOut index catalog item. Can be `store`, `aisle`, `shelf`, or `product`. This field is used only by SAP Ariba Buying and Invoicing. Example: `shelf`	No
`Territory Available`	String 255	List of one or more ISO country or region codes. SAP Ariba recommends that you use ISO 3166-1 and 3166-2 codes for this field. If there is more than one code, separate them with commas and quote the entire field. This string is case-insensitive. Example: `"US,GB,IT"`	No
`Supplier Part Auxiliary ID`	String 255	Auxiliary string. Part of the unique ID for an item, when combined with `Supplier ID` and `Supplier Part ID`. For more information about using this field for internationalization, see Catalog Internationalization and Localization [page 147]. SAP Ariba solutions support up to 255 characters. This string is case-insensitive. Never use the characters `?`, `{`, and `}` in `Supplier Part Auxiliary ID`s, as they cause validation errors. Example: `Red`	No

Field Name	Type and Length	Description	CMS Required?
`Image`	String 255	URL or pathname to a full-size image. Images must be in JPEG, GIF, PNG, or BMP format.	No
`Thumbnail`	String 255	URL or pathname to a thumbnail image, Images must be in JPEG, GIF, PNG, or BMP format.	No
`Delete`	Boolean	Key that indicates that this item is to be deleted when used an incremental catalog. The default value is `False`. This value is case-insensitive. Example: `True`	No
`IsPartial`	flag	Set to `yes` to indicate that the item can be partially specified in purchase orders.	No
`isInternalPartID`	flag	Set to `yes` to indicate that the customer catalog item is a non-standard item.	No
`IsCollection`	flag	Set to `yes` to allow the attribute to contain repeating entries. For examples of syntax for collections such as enumeration values, see Enumeration Values [page 55].	No
`DefaultRelevance`	Integer	Partially determines the relevance ranking the procurement application uses to display customer catalog items in a search. For example, `1000` is most relevant, and `0` is least relevant. Other factors that determine relevance ranking include free text relevance as determined by the SAP Ariba solution's customer catalog search engine, the Boolean attribute indicating whether a supplier is preferred, the popularity of the item across all users in the realm, and the item's line number in the customer catalog file. For an example, see Relevance Ranking Example [page 101].	No
`InKitOnly`	Boolean	Designates an item that can only be added to a cart if it is part of a catalog kit.	No
`green`	Boolean	Set to `Yes` to indicate that this item is classified as a green item.	No
`hazardousmaterials`	Boolean	Set to `Yes` to indicate that this item is classified as a hazardous material.	No
`Keywords`	String 255	Keywords for customer catalog search.	No

Field Name	Type and Length	Description	CMS Required?
`PriceConfiguration`	String 255	Defines different levels of pricing for items based on: • different quantities and dates (validity period) [page 37]. • different values of a mapped entity configured as a price lookup-key [page 39]. Every different combination of these values can also define a different price.	No
`MinimumQuantity`	Integer	Specifies the minimum order quantity for an item. Users cannot order less than the specified quantity.	Yes
`QuantityInterval`	Integer	Specifies the quantity interval for an item. The quantity that users order must match the interval. For example, if an item has a quantity interval of 2, users can order it in quantities of 2, 4, 6, 8, and so forth. Items can have both a minimum order quantity and a quantity interval; for example, users might need to order at least 2 of an item, and only be able to order it in quantities of 2, 4, 6, 8, and so forth.	Yes
`Content Unit`	String 32	Specifies the unit of a item within the item's order unit. For example, in a case of wine consisting of 12 bottles each - Case is the order unit and bottle indicates the content unit.	Yes
`Packing Quantity`	Integer	Specifies the number of content units that are packed in to an order unit. For example, if wine can be ordered in cases consisting of 12 bottles each - Case is the order unit, bottle is the content unit, and 12 is the packing quantity..	Yes
`Item Spec` [page 22]	String Field can contain upto a maximum of 6000 characters while each key-value can contain upto 255 characters.	Specifies the additional attributes or specifications of an item. You define additional attribute data as lists of key-value pairs. The syntax is `"{key1=value1;key2=value2;key3=value3;keyn=valuen}"`.	Yes

Field Name	Type and Length	Description	CMS Required?

`RelatedItems`	String
	255
		Associates other items as accessories, spare parts, follow-up, or similar items to the main items.
		You can specify multiple related items as a hash table of [“key-value”] pairs. The syntax is [{"key"="value";"key"="value"};{ * }]. The string within braces ({}) indicates one item. For example:
		`[{"type"="accessories";"artidto"="z2_tripod" };`
		`{"type"="mandatory";"artidto"="z2_case"};`
		`{"type"="followup";"artidto"="z3_case"}]`
		See, Related Items and Documents in Customer Catalogs [page 55].
			Yes

`Attachments`	String 255	Associates related documents to the customer catalog item as attachments. You can specify multiple related documents as a hash table of [“key-value”] pairs. The syntax is [{"key"="value";"key"="value";"key"="value" * };{ * }]. The string within braces ({}) indicates one document. For an example, see Related Items and Documents in Customer Catalogs [page 55].	Yes
`AuxilaryImages`	String 255	Specifies the URL or path of additional detailed images associated with the item. For more information, see Multiple Images in CIF Catalogs [page 72].	Yes
`AuxiliaryThumbnails`	String 255	Specifies the URL or path of additional thumbnail images associated with the item. For more information, see Multiple Images in CIF Catalogs [page 72].	Yes

You can also add the advanced pricing fields to the CIF and cXML catalog templates. These fields are used only when the commodities and supplier part numbers are ordered by quantity. For more information see,Advanced pricing details and fields [page 32].

Customer catalogs can contain additional supplemental attribute fields. For more information, see Supplemental Attributes [page 47].

Special Field Syntax Rules for CIF Catalogs

CIF supports specifying multiple attribute values for product commodity codes, parametric search data, and price segments using hash tables. It also supports syntax for collections (groups of repeating attributes).

Specifying Multiple Attribute Values

A hash table is a string that specifies a list of key-value pairs that are typically used to retrieve objects faster. If you are using CIF, you can use hash tables to list multiple product commodity codes, parametric search data, and price segments.

The syntax for hash tables is shown here:

```
{ 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 in 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 in the field, quote the entire field:

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

- If a quote (`"`) appears in a quoted field, escape it with an additional quote:

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

- If the characters (`;`) ,(`,`), (`}`), (`"`), or (`\`) appear in 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\"";}
  ```

Specifying Collections

A collection is a group of repeating entries in an attribute specification. SAP Ariba solutions support the syntax for collections in supplemental and parametric attributes in CIF catalogs.

> **Note**
>
> You must specify the `isCollection` property in the CIF header and set it to `yes` if you want all of the values in the collection to appear in the customer catalog. For more information about the `isCollection` property, see Type Attributes [page 160].

The syntax for collections is `"(Value1,Value2,Value3)"`.

For example:

```
DATA
7,S1,,Men’s Button Down
Shirt,32323238,15,ea,,en,,,,"(en_US,fr_FR)",,shirts,"{collared=true;color=(Red,Blue,
Green);size=(Large);}"
7,S2,,Men’s
Tshirt,32323238,25,ea,,en,,,,"(en_US,fr_FR)",,shirts,"{collared=false;color=(Blue,Gr
een);size=(Medium,Large);}"
7,S3,,Men’s Turtle Neck
```

Related Information

Required and Optional CIF Body Row Fields [page 87]

Supplier IDs in Customer Catalogs

Supplier IDs uniquely identify suppliers. Most customer catalogs use D-U-N-S® (Data Universal Numbering System) or **Ariba Network ID** (NetworkID) numbers for Supplier IDs. You declare which identification system the customer catalog uses, and for each customer catalog item, you identify the supplier that provides that item.

Supplier ID Domains

The Supplier ID Domain is the identification system you use to identify suppliers in your customer catalogs. You can use any identification system, such as D-U-N-S, or customer-specific Supplier IDs.

For easiest integration with procurement applications, use the same Supplier ID Domain that SAP Business Network displays when it lists your account information in CIF and cXML catalogs. For those customer catalogs, domains can also be NetworkIDs.

In BMEcat customer catalogs, the Supplier ID Domain is defined by the supplier ID type, which can be D-U-N-S, ILN, customer-specific, or supplier-specific Supplier IDs.

Supplier IDs in SAP Ariba Solutions

Each SAP Ariba solution maintains a list of known suppliers. During customer catalog import, they match catalog items with a supplier; any items from unknown suppliers produce a warning.

In multi-supplier CIF catalogs, an invalid supplier produces an error and must be fixed and imported again before activating the customer catalog. For more information about multi-supplier CIF catalogs, see Multi-Supplier Catalogs [page 17].

Alternatively, SAP Ariba solutions can query SAP Business Network for the profiles and IDs of new suppliers.

If you distribute your customer catalogs through SAP Business Network, the supplier list within SAP Ariba solutions contains your own Supplier ID. However, if you send customer catalogs through email or FTP, SAP Business Network might not have profiles for your organization, so administrators of SAP Ariba solutions would need to manually enter your information.

> **Note**
>
> Enable the `Application.OrganicGrowth.OrganicGrowthEnabled` parameter for SAP Ariba Buying and Invoicing to automatically generate the following supplier IDs for unknown users who punch in to SAP Ariba Catalog. This occurs when a copy order is received from the SAP Business Network and no matching users who punch in to SAP Ariba Catalog exist in SAP Ariba Buying and Invoicing based on the credentials provided in the purchase order cXML.
>
> - For common users, SAP Ariba Buying and Invoicing creates `Buyer_ System ID`.
> - For partitioned users, SAP Ariba Buying and Invoicing creates `SU_INTERNAL Vendor ID`.
>
> Contact SAP Ariba Support to enable this parameter.

Multi-Supplier Customer Catalogs

A multi-supplier customer catalog is a CIF catalog file that contains items from more than one supplier. Multi-supplier customer catalogs are not supported in cXML or BMEcat.

Manually imported local customer catalogs can contain multiple suppliers if a checkbox in Ariba Administrator is checked during import. SAP Business Network subscriptions (which are automatically loaded into SAP Ariba solutions when a catalog is updated) can contain CIF catalogs with multiple suppliers if the parameter `System.Catalog.Network.DefaultMSCif` is set in SAP Ariba solutions. If you are a supplier and want to publish multi-supplier CIF catalogs on SAP Business Network for customer subscriptions, make sure your customers have this parameter set.

> **Note**
>
> For information about using multi-supplier CIF catalogs with SAP Ariba solutions, see the *Catalog administration guide for buyers*.

The following is an example of a multi-supplier CIF catalog:

```
CIF_I_V3.0
CODEFORMAT: UNSPSC
LOADMODE: F
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
ITEMCOUNT: 3
DATA
29,RAY19844,824-2,"AA Rechargable Batteries 2-PK",26111701,3.02,BOX,0,,,,
599238,66056,66056,"Large Hot/Cold Drinking Cups",99998050,2.95,DZ,2,,,,
98-747-7712,Burgundy-1,Burgundy-1,"1996 Pinot Noir",50202203,25,EA,1,,,,
ENDOFDATA
```

Multi-supplier CIF catalogs are supported by Ariba Buyer 8.2 (and later) and SAP Ariba Procurement solutions. Multi-supplier cXML catalogs are used only in PunchOut scenarios. PunchOut shopping carts (cXML `PunchOutOrderMessage` documents) can refer to multiple suppliers. Catalog aggregators who want to use cXML should consider creating PunchOut sites instead of static customer catalog files. For more information about PunchOut, see PunchOut Catalogs [page 8].

Supplier Part Auxiliary IDs

You must uniquely identify all customer catalog items to prevent conflicts. SAP Ariba solutions identify items by using a combination of their `Supplier ID`, `Supplier Part ID`, and `Supplier Part Auxiliary ID`.

`Supplier Part Auxiliary ID`s are used in the following cases:

- To uniquely identify services, which sometimes do not have part numbers.
- To provide customer catalog items in multiple languages. For example, an item that is available in Italy, France, and Germany can have three `Supplier Part Auxiliary ID`s, which creates three customer catalog items. Each customer catalog item can have a different `Description` field (in Italian, French, or German.) Similarly, each item can have a different `Currency`, `Unit Price`, `Lead Time`, and `Supplier URL`.
  For the `Supplier Part Auxiliary ID`s, use Java locale codes, for example `it_IT`, `fr_FR`, and `de_DE`. For a list of these codes, see:
  http://java.sun.com/j2se/1.4.2/docs/guide/intl/locale.doc.html
  Buying organizations can configure SAP Ariba solutions to determine which customer catalog items to display based on the locations of users. For more information, see Creating Multilingual Catalogs [page 148].
- To differentiate items that have the same `Supplier Part Number`, but different lead times or delivery options, you can use the values `ground`, `2-day`, and `overnight` as `Supplier Part Auxiliary ID`s. (You can also specify a different price for each item.)
- To differentiate items that have the same `Supplier Part Number`, but are packaged differently, 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
  BecauseSAP Ariba solutions do not use `Descriptions` to differentiate items, they interpret these items as duplicates. To prevent duplicate items, assign them `Supplier Part Auxiliary ID`s:
  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
  Both items are then considered unique because they have different `Supplier Part Auxiliary ID`s.

> **Note**
>
> Do not use the characters `?`, `{`, and `}` in `Supplier Part Auxiliary ID`s; they cause validation errors.

SAP Ariba solutions enforce a character limit of 255 for the `Supplier Part Auxiliary ID` field.

CIF Catalog Examples

For illustrative purpose examples of CIF file include an example of using the minimum number of required fields, using all 12 basic fields, a PunchOut index catalog, a multinational customer catalog, and a parametric search data example.

Minimum Number of Fields

The CIF example of using the minimum number of fields illustrates how to omit data for fields you do not need while maintaining the correct order of all 12 required 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
```

Example of Using all 12 Basic Fields

The CIF example of using all 12 required basic fields shows a three-item CIF file.

```
CIF_I_V3.0
LOADMODE: F
CODEFORMAT: UNSPSC
CURRENCY: USD
SUPPLIERID_DOMAIN: DUNS
ITEMCOUNT: 3
TIMESTAMP: 2005-05-15 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/indepth.htm,400.00
942888710,56A12,C12,"Eames Ottoman, Blk Leather",11116767,100.00,EA,3,"Fast
MFG",http://www.acme.com/56A12.htm,http://www.mfg.com/C12/indepth.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/indepth.htm,25.95
ENDOFDATA
```

Example of PunchOut Index Catalog

The CIF example of a PunchOut index catalog illustrates how to specify PunchOut enablement in the CIF header, and how PunchOut-specific information appears in a body row field.

This example uses the `PunchOutLevel` field, which is used only by SAP Ariba Buying and Invoicing.

```
CIF_I_V3.0
LOADMODE: F
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, PunchOutLevel
CURRENCY: USD
DATA
762311901,A2C-311F,C-311F,"Configurable Chairs from Work Chairs,
Inc.",11116767,,,,,https://www.workchairs.com/configurator.asp,,,t,shelf
ENDOFDATA
```

> **Note**
>
> SAP Business Network no longer uses URLs specified within cXML `PunchOutSetupRequest` documents, which come from PunchOut catalogs. For more information, see PunchOut URL [page 29].

Multinational Example

The multinational example shows a CIF file that lists an item in three languages with different currencies. It also specifies the UNSPSC version for SAP Ariba Procurement solutions.

```
CIF_I_V3.0
LOADMODE: F
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: 2005-05-15 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 Search Data Example

The parametric example shows a two-item CIF file, with multiple product commodity 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: 2005-01-18 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,2005-12-01,2005-03-01,
{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,2005-12-01,2005-03-01,{ACME=8BE101},{TYPE="IBM PC";SPEED="66
MHZ";},COMPUTERS
ENDOFDATA
```

The following table lists the first customer catalog item, in column format 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	2005-12-01	2005-01-01	{ACME=8BE101}	{TYPE="IBM PC";SPEED="33 MHZ";}	COMPUTERS

For more information about parametric data, see Parametric Search Data [page 48].

Relevance Ranking Example

The relevance ranking example illustrates a five-item CIF file, with relevance ranking information. Use relevance tools only when instructed by your buyer customer.

```
CIF_I_V3.0
LOADMODE: F
SUPPLIERID_DOMAIN: buyersystemid
COMMENTS: Generated on Thu Sep 06 23:02:39 PDT 2007
UNUOM: FALSE
```

```
CURRENCY: USD
CHARSET: UTF-8
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, Language, Classification Codes,
DefaultRelevance
DATA
sid496,SC-611955,TNR-611955,”JCN #800 Bright Red
Toner”,,12.97,EA,1,SCHEN TONER CORPORATION,http://faisonopc.net/ariba/
R0002044.htm,,19.95,USD,en_US,{UNSPSC=55101501;},800
sid496,SC-696071,TNR-696071,”JCN #500 Orange Toner”,,20.97,EA,1,SCHEN
TONER CORPORATION,http://faisonopc.net/ariba/MAMM0137.htm,,29.95,USD,en_US,
{UNSPSC=55101501;},500
sid496,SC-698007,TNR-698007,”JCN #300 Blue Toner”,,19.47,EA,1,SCHEN
TONER CORPORATION,http://faisonopc.net/ariba/MAMM0146.htm,,24.95,USD,en_US,
{UNSPSC=55101501;},300
sid496,SC-698015,TNR-698015,”JCN #1 Green Toner”,,18.47,EA,1,SCHEN
TONER CORPORATION,http://faisonopc.net/ariba/MAMM0146.htm,,24.95,USD,en_US,
{UNSPSC=55101501;},1
sid496,SC-698260,TNR-698260,”JCN #100 Light Green
Toner”,,102.5,EA,1,SCHEN TONER CORPORATION,http://faisonopc.net/ariba/
R0001655.htm,,155,USD,en_US,{UNSPSC=55101501;},10
ENDOFDATA
```

For more information about relevance ranking, see Ranking Items by Relevance [page 53].

CIF File Editing

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 customer 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.

Related Information

Saving an Excel file to CSV Format for Import to CIF [page 103]
Opening a CIF file in Excel for Editing [page 103]

Saving an Excel file to CSV Format for Import to CIF

Use this procedure to save an Excel file to a CSV file so that it can be imported as a CIF file.

Procedure

1. In Excel, choose **File** > **Save As**.
2. Select CSV as the file type.
3. Enter an intuitive filename. The recommended name format is `suppliername_buyername`, for example `workchairs_acme`.
4. Click **Save**.
5. Excel adds extraneous commas in the header and trailer of CIF files. Open the file in Notepad or Wordpad, delete the commas from the header and trailer, and make sure that leading or trailing zeroes are not deleted.
6. Click **File** > **Save As**, change the file extension from .csv to .cif and enter double quotation marks around the filename. For example, `"workchairs_acme.cif"` The quotation marks force the file to be saved as a CIF file rather than as a TXT file.

   Do not save catalog files in raw Unicode format. For more information, see Character Encoding [page 154].
7. Click **Save**.

Opening a CIF file in Excel for Editing

Use this procedure to open a CIF file in Excel for editing.

Procedure

1. In Excel, choose **File** > **Open** and select 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 in the Data preview pane (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 choose **Format** > **Column** > **AutoFit Selection**.

7. To calculate the number of customer catalog items for the `ITEMCOUNT` field, subtract the number of the last line of data (the last customer catalog item) from the number of the first line of data (the first customer catalog item), and add one.

Related Information

Saving an Excel file to CSV Format for Import to CIF [page 103]

cXML Catalogs

cXML Basics [page 105]

cXML Catalog Files [page 106]

cXML Catalog Examples [page 111]

cXML Basics

You can create customer catalogs using commerce eXtensible Markup Language (cXML). cXML is an XML-based language and follows standard XML syntax.

XML is a text markup language for interchange of structured data. It lets designers create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

For more information about XML, see the following websites:

http://www.xml.com

http://www.xml.org

cXML, a language for conveying data related to e-commerce, is an open standard promoted by SAP 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. Because cXML is an XML-based language, you can use a variety of applications to generate and parse it. cXML is the ideal format for automatically generated customer catalogs, and it supports all the customer catalog features of CIF.

The description of cXML here is appropriate for suppliers who create content for SAP Business Network and for customer catalog administrators working with procurement applications.

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: standard catalogs, and contract files. You can use any standard ASCII characters in the catalog filename. Use the extension `.xml` for cXML catalogs.

Customer catalog files list customer catalog items, their descriptions, and their prices. Customer catalog files include one or more supplier IDs to identify the supplier. However, each customer catalog can reference only one supplier; multi-supplier customer catalogs are not supported in cXML

Contract files list only prices for items. Each item in these files refers to an item in a catalog file. Distributing pricing information in a separate file lets you manage pre-negotiated, customer-specific contract pricing. Your SAP Ariba

solution does not support the use of cXML contract files. To offer special contract pricing for your customers, use SAP Ariba Contracts. For more information about contracts, see Managing Contract Compliance.

Related Information

cXML Catalog Files [page 106]

cXML Catalog Files

Customer catalog files list items, their descriptions, and prices. Every cXML file has two sections, a header, and a body.

- The header specifies general attributes that apply to all the data in the file.
- The body describes the items in the customer catalog.

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>
```

Basic cXML syntax

cXML follows standard XML syntax rules, and adds support for e-commerce-specific notation.

- 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. 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>
  ```

  does 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 customer catalog item, for example:

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

  Customers can configure their procurement applications to display the appropriate customer catalog item, based on the locations of users. For more information about multilingual customer catalogs, see Catalog Internationalization and Localization [page 147].
  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/j2se/1.4.2/docs/guide/intl/locale.doc.html
  ```

Related Information

cXML Headers in Customer Catalog Files [page 107]
cXML Catalog Body [page 108]
cXML Catalog Examples [page 111]

cXML Headers in Customer Catalog Files

A cXML header contains an XML version declaration and a DOCTYPE specification.

The following table describes cXML header elements:

`?xml`	Specifies the XML version and character encoding. `"UTF-8"` (Universal Transformation Format) supports all character encodings currently supported by SAP Ariba solutions. You can also use `"ISO 8859-1"`, which corresponds to Western European language characters (Latin 1). For more information, see Catalog Character Set and Character Encoding [page 153].
`!DOCTYPE`	Specifies the type of cXML file and the URL for the cXML DTD. The DTD describes the format and syntax of all types of cXML documents and is used by applications to validate files.

> **Note**
>
> Verify that your `UTF-8` catalog files do not contain a byte-order mark. Byte-order marks are conventionally used as a marker to indicate that text is encoded in UTF-8 files. Notepad can add byte-order marks automatically to UTF-8 files and they can cause errors in SAP Ariba solutions.

cXML Load Mode Attribute

Starting with cXML 1.2.008, catalogs can use a `loadmode` attribute to indicate whether they are full or incremental.

In catalogs used by SAP Ariba solutions, use `<Index loadmode="Incremental">` to indicate incremental catalogs and `<Index loadmode="Full">` to indicate full catalogs. If the `loadmode` attribute is missing from the customer catalog file, the SAP Ariba solution assumes that it is a full load and generates a warning.

> **Note**
>
> Ariba buyer 8.0 and 8.1 support only full customer catalogs. They interpret all customer catalogs as full and if customer catalogs use the `Index loadmode` attribute, it must be set to `Full`. Ariba Buyer version 8.2 and later support incremental customer catalogs.

cXML Catalog Body

The body of customer catalog files contains one or more `IndexItem` elements of type `<IndexItemAdd>`, `<IndexItemDelete>`, or `<IndexItemPunchout>`.

The following example shows a typical customer catalog file structure:

```
header information
<Index loadmode="Full">
    <SupplierID domain="DUNS">942888710</SupplierID>
    <Comments xml:lang="en-US">Sample cXML/Index</Comments>
    <SearchGroup>
        used only by Ariba Buyer 7.0 and earlier
        ...
    </SearchGroup>
    <IndexItem>
        ...
    </IndexItem>
</Index>
```

IndexItemAdd

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.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index loadmode="Full">
    <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-US">
                    <ShortName>Men's Shoes</ShortName>
                    Black shoes with velcro clasp
                </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.

IndexItemDelete

The `IndexItemDelete` element 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.

The following example shows an incremental customer catalog using the `IndexItemDelete` element.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.007/cXML.dtd">
<Index loadmode="Incremental">
    <SupplierID domain="duns">771231234</SupplierID>
    <IndexItem>
        <IndexItemDelete>
            <ItemID>
                <SupplierPartID>pn12356</SupplierPartID>
                <SupplierPartAuxiliaryID>pw-123</SupplierPartAuxiliaryID>
            </ItemID>
        </IndexItemDelete>
        <IndexItemDelete>
            <ItemID>
```

```
                <SupplierPartID>pn12357</SupplierPartID>
            </ItemID>
        </IndexItemDelete>
    </IndexItem>
</Index>
```

> **Note**
>
> Ariba Buyer 8.0 and 8.1 accept only full customer catalog loading. They interpret all catalogs as full and if customer catalogs use the `Index loadmode` attribute, it must be set to `Full`. Ariba Buyer versions 8.2 and later support incremental customer catalogs.

IndexItemPunchout

As its name implies, the `IndexItemPunchout` element identifies a PunchOut catalog item.

The following example illustrates a `IndexItemPunchout` element:

```
<IndexItem>
    <IndexItemPunchout>
        <ItemID>
            <SupplierPartID>pn12399</SupplierPartID>
        </ItemID>
        <PunchoutDetail punchoutLevel="shelf">
            <Description xml:lang="en-US">RubySlippers</Description>
            <URL>http://oz.com/Dorothy/shoes/configurator.asp</URL>
            <Classification domain="UNSPSC">5136030000</Classification>
            <ManufacturerName>WizardShoes</ManufacturerName>
            <ManufacturerPartID>WSRS1</ManufacturerPartID>
            <ExpirationDate>2010-01-01</ExpirationDate>
            <EffectiveDate>2005-01-24</EffectiveDate>
            <SearchGroupData>
                <Name xml:lang="en-US">"DRESSSHOES"</Name>
                <SearchDataElement name="SIZE" value="4"/>
                <SearchDataElement name="COLOR" value="RUBY RED"/>
                <SearchDataElement name="GENDER" value="WOMENS"/>
            </SearchGroupData>
            <TerritoryAvailable>US</TerritoryAvailable>
            <TerritoryAvailable>GB</TerritoryAvailable>
        </PunchoutDetail>
    </IndexItemPunchout>
</IndexItem>
```

If the buying organization uses SAP Ariba Buying and Invoicing, suppliers can add the `punchoutLevel` attribute to specify how SAP Ariba Buying and Invoicing should display the PunchOut item. For more information, see punchoutLevel Attribute [page 30].

cXML Catalog Examples

Two cXML catalog examples illustrate a cXML catalog file containing one add item, and a cXML catalog file containing one PunchOut item.

Add item example

The example cXML catalog file that illustrates the `IndexItemAdd` element also illustrates the use of parametric search data and specifies a UNSPSC version for use by an SAP Ariba solution.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index loadmode="Full">
    <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>
            <ItemDetail>
                <UnitPrice>
                    <Money currency="USD">60</Money>
                </UnitPrice>
                <Description xml:lang="en-US">
                    <ShortName>Mens Shoes</ShortName>
                    Black shoes with velcro clasp
                </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>2005-06-01</ExpirationDate>
                <EffectiveDate>2004-01-25</EffectiveDate>
                <SearchGroupData>
                    <Name xml:lang="en-US">"DRESSSHOE"</Name>
                    <SearchDataElement name="SIZE" value="12"/>
                    <SearchDataElement name="COLOR" value="BLACK"/>
                    <SearchDataElement name="GENDER" value="MENS"/>
                </SearchGroupData>
                <TerritoryAvailable>US</TerritoryAvailable>
                <TerritoryAvailable>GB</TerritoryAvailable>
            </IndexItemDetail>
        </IndexItemAdd>
    </IndexItem>
</Index>
```

For information about parametric search data, see Parametric Search Data [page 48].

PunchOut index catalog example

The example cXML catalog file that illustrates the `IndexItemPunchout` element provides multiple examples of PunchOut items.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.012/cXML.dtd">
<Index>
    <SupplierID domain="duns">611429481</SupplierID>
    <Comments xml:lang="en-US">
        Sample cXML/Index
    </Comments>
    <IndexItem>
        <IndexItemPunchout>
            <ItemID>
                <SupplierPartID>1-57231-805-8</SupplierPartID>
            </ItemID>
            <PunchoutDetail punchoutLevel="product">
                <Description xml:lang="en-US">Whiteboard markers, one dozen</
Description>
                <URL>http://www.whitebd.com/cXML/PunchoutSetup/Punchoutshop.asp</
URL>
                <Classification domain="UNSPSC">55101524</Classification>
            </PunchoutDetail>
        </IndexItemPunchout>
    </IndexItem>
    <IndexItem>
        <IndexItemPunchout>
            <ItemID>
                <SupplierPartID>VTS-4976-200</SupplierPartID>
            </ItemID>
            <PunchoutDetail punchoutLevel="product">
                <Description xml:lang="en-US">Whiteboard Eraser, felt</Description>
                <URL>http://www.whitebd.com/cXML/PunchoutSetup/Punchoutshop.asp</
URL>
                <Classification domain="UNSPSC">43232005</Classification>
            </PunchoutDetail>
        </IndexItemPunchout>
    </IndexItem>
    <IndexItem>
        <IndexItemPunchout>
            <ItemID>
                <SupplierPartID>GS3600</SupplierPartID>
            </ItemID>
            <PunchoutDetail punchoutLevel="product">
                <Description xml:lang="en-US">Whiteboard Cleaner, non-toxic</
Description>
                <URL>http://www.whitebd.com/cXML/PunchoutSetup/Punchoutshop.asp</
URL>
                <Classification domain="UNSPSC">52161512</Classification>
            </PunchoutDetail>
        </IndexItemPunchout>
    </IndexItem>
</Index>
```

BMEcat Catalogs

BMEcat and Catalog Example [page 113]

Supported BMEcat Elements and Attributes [page 122]

BMEcat and Catalog Example

BMEcat, a format for conveying catalog data, is an open standard developed and maintained by Bundesverband Materialwirtschaft, Einkauf und Logistik e. V., the German association for materials management, purchasing, and logistics. BMEcat is an extension of XML.

BMEcat

The BMEcat standard supports the following features:

- Tiered pricing
- Multiple images and thumbnails
- Related documents such as product information sheets or user manuals
- Related items, including similar items, mandatory items, and optional accessories and spare parts
- Minimum order quantities and quantity intervals
- Price flags that indicate whether duty, insurance, freight, or packing are included in the price of an item
- Date-dependent pricing
- Defined keywords for search

Because BMEcat is an XML-based standard, you can use a variety of applications to generate it. The description of cXML here is appropriate for suppliers who create content for buyers who order through SAP Ariba solutions and for customer catalog administrators working with procurement applications. For BMEcat news and DTDs (Document Type Definitions), visit visit http://www.bmecat.org/ .

Supported Versions

SAP Ariba solutions support BMEcat 1.2. They do not support BMEcat 2005.

Limitations

At this time, BMEcat catalog items can only be added to requisitions. They cannot be used in contracts or invoices.

You cannot add BMEcat items to catalog kits. However, since BMEcat catalogs support related items, main items with mandatory items function in the same way as kits do by grouping items together.

At this time, SAP Ariba solutions support major BMEcat catalog functionality, but it do not support the full DTD. Unsupported elements, attributes, and values are ignored during customer catalog import. Extrinsics are not supported.

Filenames

You can use any standard ASCII characters in the catalog filename. Use the extension `.xml` for cXML catalogs.

BMEcat syntax

Customer catalog files list items, their descriptions, and prices. Every BMEcat file has two sections, a header, and a body.

- The header specifies general attributes that apply to all the data in the file.
- The body describes the items in the customer catalog.

BMEcat catalog files have the following basic format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BMECAT SYSTEM "bmecat_new_catalog_1_2.dtd">
<BMECAT version="1.2">
```

BMEcat catalog example

The following example customer catalog file adds a new item with tiered pricing, multiple images, related documents, and a set of related items that are added in the same customer catalog.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BMECAT SYSTEM "bmecat_new_catalog_1_2.dtd">
<BMECAT version="1.2">
    <HEADER>
        <GENERATOR_INFO>PSX Toolset v. 1.5</GENERATOR_INFO>
        <CATALOG>
            <LANGUAGE>eng</LANGUAGE>
      <CATALOG_ID>SAMPLECAT_1</CATALOG_ID>
      <CATALOG_VERSION>001.001</CATALOG_VERSION>
      <CATALOG_NAME>Sample Catalog</CATALOG_NAME>
      <DATETIME type="generation_date">
                <DATE>2015-11-13</DATE>
                <TIME>09:41:25</TIME>
            </DATETIME>
            <TERRITORY>US</TERRITORY>
            <CURRENCY>USD</CURRENCY>
            <MIME_ROOT>.</MIME_ROOT>
        </CATALOG>
        <BUYER>
            <BUYER_ID type="buyer_specific">408000074</BUYER_ID>
            <BUYER_NAME>Buyer ABC</BUYER_NAME>
        </BUYER>
                <AGREEMENT>
                    <AGREEMENT_ID>FLN 12/03</AGREEMENT_ID>
                    <DATETIME type="agreement_start_date">
                        <DATE>2015-12-01</DATE>
                    </DATETIME>
                    <DATETIME type="agreement_end_date">
                        <DATE>2018-12-31</DATE>
                    </DATETIME>
        </AGREEMENT>
        <SUPPLIER>
      <SUPPLIER_ID type="buyer_specific">sid499</SUPPLIER_ID>
      <SUPPLIER_ID type="supplier_specific">Test Company</SUPPLIER_ID>
      <SUPPLIER_NAME>Test Supplier</SUPPLIER_NAME>
        </SUPPLIER>
    </HEADER>
    <T_NEW_CATALOG>
        <ARTICLE mode="new">
            <SUPPLIER_AID>A1-00</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Smartphone</DESCRIPTION_SHORT>
```

```
                <DESCRIPTION_LONG>Smartphone: Expandable up to 64 GB. Brilliant 6"
screen.</DESCRIPTION_LONG>
                <BUYER_AID type="BUYER" />
                <DELIVERY_TIME>21</DELIVERY_TIME>
                <KEYWORD>Mobile</KEYWORD>
                <KEYWORD>SMARTPHONE</KEYWORD>
                <ARTICLE_STATUS type="core_article">true</ARTICLE_STATUS>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>10</QUANTITY_MIN>
                <QUANTITY_INTERVAL>5</QUANTITY_INTERVAL>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
                <DATETIME type="valid_end_date">
                    <DATE>2015-12-25</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>149</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                    <LOWER_BOUND>10</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>140</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                    <LOWER_BOUND>20</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>135</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                    <LOWER_BOUND>30</LOWER_BOUND>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
            <MIME_INFO>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/main.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone main image</MIME_DESCR>
                    <MIME_ALT>Smartphone main image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>1</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/moto_aux1.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone side left image</MIME_DESCR>
                    <MIME_ALT>Smartphone side left image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>2</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/moto_aux2.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone side right image</MIME_DESCR>
```

```
                    <MIME_ALT>Smartphone side right image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>3</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/moto_aux3.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone backside image</MIME_DESCR>
                    <MIME_ALT>Smartphone backside image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>4</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>application/pdf</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/
smartphone_guidebook.pdf</MIME_SOURCE>
                    <MIME_DESCR>Smartphone User Manual</MIME_DESCR>
                    <MIME_ALT>Smartphone User Manual</MIME_ALT>
                    <MIME_PURPOSE>data_sheet</MIME_PURPOSE>
                    <MIME_ORDER>5</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>url</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/
smartphone_specs.html</MIME_SOURCE>
                    <MIME_DESCR>Smartphone Product Specifications</MIME_DESCR>
                    <MIME_ALT>Smartphone Product Specifications</MIME_ALT>
                    <MIME_PURPOSE>data_sheet</MIME_PURPOSE>
                    <MIME_ORDER>6</MIME_ORDER>
                </MIME>
            </MIME_INFO>
            <ARTICLE_REFERENCE type="accessories">
                <ART_ID_TO>A1-03</ART_ID_TO>
            </ARTICLE_REFERENCE>

            <ARTICLE_REFERENCE type="mandatory">
                <ART_ID_TO>A1-01</ART_ID_TO>

            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="mandatory">
                <ART_ID_TO>A1-0</ART_ID_TO>
            </ARTICLE_REFERENCE>
        </ARTICLE>
        <ARTICLE mode="new">
            <SUPPLIER_AID>A1-01</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Wireless Charger for Smartphones/Tablets</
DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Wireless Charger for Smartphones/Tablets</
DESCRIPTION_LONG>
                <DELIVERY_TIME>14</DELIVERY_TIME>
                <KEYWORD>Charger</KEYWORD>
                <KEYWORD>Wireless</KEYWORD>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>10</QUANTITY_MIN>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
```

```
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>49.90</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
        </ARTICLE>

        <ARTICLE mode="new">
            <SUPPLIER_AID>A1-02</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Micro USB Data Cable</DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Micro USB Data Cable</DESCRIPTION_LONG>
                <DELIVERY_TIME>14</DELIVERY_TIME>
                <KEYWORD>USB</KEYWORD>
                <KEYWORD>Cable</KEYWORD>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>10</QUANTITY_MIN>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>5</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
        </ARTICLE>

        <ARTICLE mode="new">
            <SUPPLIER_AID>A1-03</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Smartphone Case, Gold Slim-Fit</
DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Smartphone Case, Gold Slim-Fit Smart Case Cover
for Smartphone</DESCRIPTION_LONG>
                <DELIVERY_TIME>14</DELIVERY_TIME>
                <KEYWORD>Case</KEYWORD>
                <KEYWORD>Cover</KEYWORD>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>1</QUANTITY_MIN>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>10</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
```

```
        </ARTICLE>


    </T_NEW_CATALOG>
</BMECAT>
```

BMEcat catalog example

The following example catalog file adds a new item with tiered pricing, multiple images, related documents, and a set of related items that are added in the same catalog.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BMECAT SYSTEM "bmecat_new_catalog_1_2.dtd">
<BMECAT version="1.2">
    <HEADER>
        <GENERATOR_INFO>PSX Toolset v. 1.5</GENERATOR_INFO>
        <CATALOG>
            <LANGUAGE>eng</LANGUAGE>
      <CATALOG_ID>SAMPLECAT_1</CATALOG_ID>
      <CATALOG_VERSION>001.001</CATALOG_VERSION>
      <CATALOG_NAME>Sample Catalog</CATALOG_NAME>
      <DATETIME type="generation_date">
                <DATE>2015-11-13</DATE>
                <TIME>09:41:25</TIME>
            </DATETIME>
            <TERRITORY>US</TERRITORY>
            <CURRENCY>USD</CURRENCY>
            <MIME_ROOT>.</MIME_ROOT>
        </CATALOG>
        <BUYER>
            <BUYER_ID type="buyer_specific">408000074</BUYER_ID>
            <BUYER_NAME>Buyer ABC</BUYER_NAME>
        </BUYER>
                <AGREEMENT>
                    <AGREEMENT_ID>FLN 12/03</AGREEMENT_ID>
                    <DATETIME type="agreement_start_date">
                        <DATE>2015-12-01</DATE>
                    </DATETIME>
                    <DATETIME type="agreement_end_date">
                        <DATE>2018-12-31</DATE>
                    </DATETIME>
        </AGREEMENT>
        <SUPPLIER>
      <SUPPLIER_ID type="buyer_specific">sid499</SUPPLIER_ID>
      <SUPPLIER_ID type="supplier_specific">Test Company</SUPPLIER_ID>
      <SUPPLIER_NAME>Test Supplier</SUPPLIER_NAME>
        </SUPPLIER>
    </HEADER>
    <T_NEW_CATALOG>
        <ARTICLE mode="new">
            <SUPPLIER_AID>A1-00</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Smartphone</DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Smartphone: Expandable up to 64 GB. Brilliant 6"
screen.</DESCRIPTION_LONG>
                <BUYER_AID type="BUYER" />
                <DELIVERY_TIME>21</DELIVERY_TIME>
                <KEYWORD>Mobile</KEYWORD>
                <KEYWORD>SMARTPHONE</KEYWORD>
                <ARTICLE_STATUS type="core_article">true</ARTICLE_STATUS>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
```

```
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>10</QUANTITY_MIN>
                <QUANTITY_INTERVAL>5</QUANTITY_INTERVAL>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
                <DATETIME type="valid_end_date">
                    <DATE>2015-12-25</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>149</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                    <LOWER_BOUND>10</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>140</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                    <LOWER_BOUND>20</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>135</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                    <LOWER_BOUND>30</LOWER_BOUND>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
            <MIME_INFO>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/main.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone main image</MIME_DESCR>
                    <MIME_ALT>Smartphone main image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>1</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/moto_aux1.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone side left image</MIME_DESCR>
                    <MIME_ALT>Smartphone side left image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>2</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/moto_aux2.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone side right image</MIME_DESCR>
                    <MIME_ALT>Smartphone side right image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>3</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/moto_aux3.jpeg</
MIME_SOURCE>
                    <MIME_DESCR>Smartphone backside image</MIME_DESCR>
```

```
                    <MIME_ALT>Smartphone backside image</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>4</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>application/pdf</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/
smartphone_guidebook.pdf</MIME_SOURCE>
                    <MIME_DESCR>Smartphone User Manual</MIME_DESCR>
                    <MIME_ALT>Smartphone User Manual</MIME_ALT>
                    <MIME_PURPOSE>data_sheet</MIME_PURPOSE>
                    <MIME_ORDER>5</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>url</MIME_TYPE>
                    <MIME_SOURCE>http://12.34.56.789:8888/myapp/
smartphone_specs.html</MIME_SOURCE>
                    <MIME_DESCR>Smartphone Product Specifications</MIME_DESCR>
                    <MIME_ALT>Smartphone Product Specifications</MIME_ALT>
                    <MIME_PURPOSE>data_sheet</MIME_PURPOSE>
                    <MIME_ORDER>6</MIME_ORDER>
                </MIME>
            </MIME_INFO>
            <ARTICLE_REFERENCE type="accessories">
                <ART_ID_TO>A1-03</ART_ID_TO>
            </ARTICLE_REFERENCE>

            <ARTICLE_REFERENCE type="mandatory">
                <ART_ID_TO>A1-01</ART_ID_TO>

            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="mandatory">
                <ART_ID_TO>A1-0</ART_ID_TO>
            </ARTICLE_REFERENCE>
        </ARTICLE>
        <ARTICLE mode="new">
            <SUPPLIER_AID>A1-01</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Wireless Charger for Smartphones/Tablets</
DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Wireless Charger for Smartphones/Tablets</
DESCRIPTION_LONG>
                <DELIVERY_TIME>14</DELIVERY_TIME>
                <KEYWORD>Charger</KEYWORD>
                <KEYWORD>Wireless</KEYWORD>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>10</QUANTITY_MIN>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>49.90</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
        </ARTICLE>

        <ARTICLE mode="new">
```

```
            <SUPPLIER_AID>A1-02</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Micro USB Data Cable</DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Micro USB Data Cable</DESCRIPTION_LONG>
                <DELIVERY_TIME>14</DELIVERY_TIME>
                <KEYWORD>USB</KEYWORD>
                <KEYWORD>Cable</KEYWORD>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>10</QUANTITY_MIN>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>5</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
        </ARTICLE>

        <ARTICLE mode="new">
            <SUPPLIER_AID>A1-03</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>Smartphone Case, Gold Slim-Fit</
DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Smartphone Case, Gold Slim-Fit Smart Case Cover
for Smartphone</DESCRIPTION_LONG>
                <DELIVERY_TIME>14</DELIVERY_TIME>
                <KEYWORD>Case</KEYWORD>
                <KEYWORD>Cover</KEYWORD>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44111501</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <QUANTITY_MIN>1</QUANTITY_MIN>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2015-12-01</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>10</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>0.19</TAX>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
        </ARTICLE>


    </T_NEW_CATALOG>
</BMECAT>
```

Supported BMEcat Elements and Attributes

SAP Ariba supports a subset of available elements and attributes in the BMEcat 1.2 standard. The customer catalog files are validated as per the BMEcat DTD standards.

> **Note**
>
> SAP Ariba requires all elements that are indicated as mandatory in the BMEcat 1.2 standard for customer catalog validation at import. However, SAP Ariba Procurement solutions do not use all of these elements, and ignore those that are not used.

Header elements and attributes

SAP Ariba supports the `CATALOG` and `SUPPLIER` elements in the `HEADER` element. It requires the mandatory `BUYER` element, but ignores it.

HEADER

BMEcat Element/Attribute	Mandatory (as per BMEcat 1.2 standard)	Type and length	Description

HEADER	Yes	-	Contains header information.
			`<HEADER>`
			`  <GENERATOR_INFO>...</GENERATOR_INFO>`
			`  <CATALOG>...</CATALOG>`
			`  <SUPPLIER>...</SUPPLIER>`
			`</HEADER>`

CATALOG	Yes	-	Information for identifying and describing the product customer catalog
			`<CATALOG>`
			`  <LANGUAGE>...</LANGUAGE>`
			`  <CATALOG_ID>...</CATALOG_ID>`
			`  <CATALOG_VERSION>...</CATALOG_VERSION>`
			`  <CATALOG_NAME>...</CATALOG_NAME>`
			`  <DATETIME type="generation_date">`
			`  <DATE>...</DATE>`
			`  <TIME>...</TIME>`
			`  </DATETIME>`
			`  <TERRITORY>...</TERRITORY>`
			`  <CURRENCY>...</CURRENCY>`
			`  <PRICE_FLAG type="...">...</PRICE_FLAG>`
			`</CATALOG>`

BMEcat Element/Attribute	Mandatory (as per BMEcat 1.2 standard)	Type and length	Description
LANGUAGE	Yes	String	Defines the language of the customer catalog data. SAP Ariba Procurement solutions copy the LANGUAGE in the header to all line items.
CATALOG_ID	Yes SAP Ariba Procurement solutions ignore this mandatory field.	String (20)	Unique ID for customer catalog identification.
CATALOG_VERSION	Yes SAP Ariba Procurement solutions ignore this mandatory field.	String (7)	SAP Ariba Procurement solutions ignore this mandatory field and uses their own versioning system for customer catalog subscriptions.
CATALOG_NAME	No	String (100)	Name that describes the customer catalog.
DATETIME	No	String (20)	Timestamp indicating the customer catalog generation date.
DATE	Yes	Date	Indicates the date.
TIME	No	Time	Indicates the time.
TERRITORY	No	String (2)	Defines the territories (region, federal state, country, continent) in which the products described in the customer catalog are available. Specify territory using ISO 3166-1 codes. SAP Ariba Procurement solutions copy the TERRITORY in the header to all line items.
CURRENCY	No	String (3)	Defines the currency of the prices of the items in the customer catalog. Specify currency with three-letter ISO 4217 currency codes. SAP Ariba Procurement solutions copy the CURRENCY in the header to all line items.

BMEcat Element/Attribute	Mandatory (as per BMEcat 1.2 standard)	Type and length	Description
`PRICE_FLAG type="..."`	No	String (50)	Indicates whether duty, insurance, freight, or packing are included in the price of an item. SAP Ariba Procurement solutions support the following PRICE_FLAG types: • `incl_freight` • `incl_packing` • `incl_assurrance` • `incl_duty`
`SUPPLIER`	Yes	-	`<HEADER> . <SUPPLIER> <SUPPLIER_ID type="buyer_specific">...</SUPPLIER_ID> <SUPPLIER_ID type="supplier_specific">...</SUPPLIER_ID> <SUPPLIER_NAME>ABC Supplies</SUPPLIER_NAME> </SUPPLIER> . </HEADER>`
`SUPPLIER_ID type="buyer_specific"`	Yes	String (50)	SAP Ariba Procurement solutions map this value to the supplier domain `buyersystemid`.
`SUPPLIER_ID type="supplier_specific"`	No	String (50)	SAP Ariba Procurement solutions map this value to the supplier's ANID.
`SUPPLIER_NAME`	Yes	String (50)	SAP Ariba Procurement solutions ignore this field and assign the supplier specified by the catalog manager during import to the catalog.

Related Information

Body elements and attributes [page 125]
BMEcat and Catalog Example [page 113]

Body elements and attributes

SAP Ariba supports the T_NEW_CATALOG and T_UPDATE_CATALOG elements for defining the body of the customer catalog.

ARTICLE

SAP Ariba supports the ARTICLE element in T_NEW_CATALOG and T_UPDATE_CATALOG elements for defining an article in the customer catalog.

Element/Attribute	Mandatory	Type and length	Notes

ARTICLE mode="new"	Yes	String	SAP Ariba supports the new value for the mode attribute.
			`<T_NEW_CATALOG>`
			`  .`
			`  <ARTICLE mode="new">`
			`  .`
			`  <SUPPLIER_AID>55-K-31</SUPPLIER_AID>`
			`  <ARTICLE_DETAILS>...</ARTICLE_DETAILS>`
			`  <ARTICLE_FEATURES>...</ARTICLE_FEATURES>`
			`  <ARTICLE_ORDER_DETAILS>...</ARTICLE_ORDER_DETAILS>`
			`  <ARTICLE_PRICE_DETAILS>...</ARTICLE_PRICE_DETAILS>`
			`  <MIME_INFO>...</MIME_INFO>`
			`  <ARTICLE_REFERENCE type="followup">...</ARTICLE_REFERENCE>`
			`  .`
			`  </ARTICLE>`
			`  .`
			`</T_NEW_CATALOG>`

SUPPLIER_AID	Yes	String (35)	Supplier's unique article number

Element/Attribute	Mandatory	Type and length	Notes

`ARTICLE_DETAILS`	Yes	-	Contains the elements that define the details of the article.
			
			`<ARTICLE>`
			`  .`
			`  <ARTICLE_DETAILS>`
			`  <DESCRIPTION_SHORT>...</DESCRIPTION_SHORT>`
			`  <DESCRIPTION_LONG>...</DESCRIPTION_LONG>`
			`  <EAN>...</EAN>`
			`  <BUYER_AID type="...">...</BUYER_AID>`
			`  <MANUFACTURER_AID>...</MANUFACTURER_AID>`
			`  <MANUFACTURER_NAME>...</MANUFACTURER_NAME>`
			`  <DELIVERY_TIME>...</DELIVERY_TIME>`
			`  <SPECIAL_TREATMENT_CLASS type="...">...</SPECIAL_TREATMENT_CLASS>`
			`  <KEYWORD>...</KEYWORD>`
			`  <ARTICLE_STATUS type="...">...</ARTICLE_STATUS>`
			`  </ARTICLE_DETAILS>`
			`  .`
			`</ARTICLE>`

`ARTICLE_FEATURES`	Yes	-	Used to define the classification of the article.
			
			`<ARTICLE>`
			`  .`
			`  <ARTICLE_FEATURES>`
			
			`<REFERENCE_FEATURE_SYSTEM_NAME>...</REFERENCE_FEATURE_SYSTEM_NAME>`
			`  <REFERENCE_FEATURE_GROUP_ID>...</REFERENCE_FEATURE_GROUP_ID>`
			`  or`
			
			`<REFERENCE_FEATURE_GROUP_NAME>...<REFERENCE_FEATURE_GROUP_NAME>`
			`  </ARTICLE_FEATURES>`
			`  .`
			`</ARTICLE>`

Element/Attribute	Mandatory	Type and length	Notes

ARTICLE_ORDER_DETAILS	Yes	-	Contains elements which define the terms and conditions that are applied when a particular article is ordered.
			
			`<ARTICLE>`
			`  .`
			`  <ARTICLE_ORDER_DETAILS>`
			`  <ORDER_UNIT>...</ORDER_UNIT>`
			`  <PRICE_QUANTITY>...</PRICE_QUANTITY>`
			`  <QUANTITY_MIN>...</QUANTITY_MIN>`
			`  <QUANTITY_INTERVAL>...</QUANTITY_INTERVAL>`
			`  </ARTICLE_ORDER_DETAILS>`
			`.`
			`</ARTICLE>`

ARTICLE_PRICE_DETAILS	Yes	-	Specifies price-related data.
			
			`<ARTICLE_PRICE_DETAILS>`
			`  <DATETIME type="valid_start_date">`
			`  <DATE>...</DATE>`
			`  </DATETIME>`
			`  <DATETIME type="valid_end_date">`
			`  <DATE>...</DATE>`
			`  </DATETIME>`
			`  <ARTICLE_PRICE price_type="net_customer">`
			`  .`
			`  </ARTICLE_PRICE>`
			`</ARTICLE_PRICE_DETAILS>`

MIME_INFO	No	-	You can include references to additional multimedia documents in an article.
			
			`<ARTICLE>`
			`  .`
			`  <MIME_INFO>`
			`  <MIME>`
			`  <MIME_TYPE>...</MIME_TYPE>`
			`  <MIME_SOURCE>...</MIME_SOURCE>`
			`  <MIME_DESCR>...</MIME_DESCR>`
			`  <MIME_ALT>...</MIME_ALT>`
			`  <MIME_PURPOSE>...</MIME_PURPOSE>`
			`  </MIME>`
			`  </MIME_INFO>`
			`  .`
			`</ARTICLE>`

Element/Attribute	Mandatory	Type and length	Notes

`ARTICLE_REFERENCE type="..."`	No	String (20)	Used to refer from one article to another.
			SAP Ariba supports the following values for `ARTICLE_REFERENCE type`:
			• `mandatory`
			• `similar`
			• `sparepart`
			• `accessories`
			• `followup`
			`<ARTICLE>`
			`.`
			`<ARTICLE_REFERENCE type="...">`
			`<ART_ID_TO>...</ART_ID_TO>`
			`</ARTICLE_REFERENCE>`
			`.`
			`</ARTICLE>`

ARTICLE_DETAILS

SAP Ariba supports the `ARTICLE_DETAILS` element within the `ARTICLE` element and consists of data fields which identify an article.

Element/Attribute	Mandatory	Type and length	Notes

`ARTICLE_DETAILS`	Yes		Contains the elements that define the details of the article.
			`<ARTICLE>`
			`.`
			`<ARTICLE_DETAILS>`
			`<DESCRIPTION_SHORT>...</DESCRIPTION_SHORT>`
			`<DESCRIPTION_LONG>...</DESCRIPTION_LONG>`
			`<EAN>...</EAN>`
			`<BUYER_AID type="...">...</BUYER_AID>`
			`<MANUFACTURER_AID>...</MANUFACTURER_AID>`
			`<MANUFACTURER_NAME>...</MANUFACTURER_NAME>`
			`<DELIVERY_TIME>...</DELIVERY_TIME>`
			`<SPECIAL_TREATMENT_CLASS type="...">...</SPECIAL_TREATMENT_CLASS>`
			`<KEYWORD>...</KEYWORD>`
			`<ARTICLE_STATUS type="...">...</ARTICLE_STATUS>`
			`</ARTICLE_DETAILS>`
			`<SEGMENT>...</SEGMENT>`
			`.`
			`</ARTICLE>`

`DESCRIPTION_SHORT`	Yes	String (80)	Short description or name of the article.
`DESCRIPTION_LONG`	No	String (2000)	Long description of the article. SAP Ariba Procurement solutions enforce a 2000 character limit.
`EAN`	No	String (14)	European article number. This field value is mapped to the new `Additional Part IDs` catalog field by default.
`BUYER_AID type=...`	Yes	String (18)	Article ID of the buying company. The type attribute specifies the type of the ID. If you use multiple types, then the values for each type must be different. This field value is mapped to the new `Additional Part IDs` catalog field by default.
`MANUFACTURER_AID`	No	String (50)	Article ID of the manufacturer.

Element/Attribute	Mandatory	Type and length	Notes
`MANUFACTURER_NAME`	No	String (50)	Name of the manufacturer.
`DELIVERY_TIME`	No	Number (6)	Time in working days needed by the supplier to supply the article.
`SPECIAL_TREATMENT_CLASS type=...`	No	String (20)	Special treatment class used for additional classification of hazardous goods or substances. The `type` attribute specifies the dangerous goods classification scheme and the value indicates the actual class within this scheme. This field value is mapped to the `SpecialTreatmentClass` catalog field by default.
`KEYWORD`	No	String (50)	SAP Ariba Procurement solutions use `KEYWORD` values as one of the factors that determines item relevance in search results.
`ARTICLE_STATUS type=...`	No	String (250)	This element is used to classify an article in terms of its special characteristics. The status type is specified by the `type` attribute and the value indicates the text description of the special characteristics. This field value is mapped to the `ItemSpecialStatus` catalog field by default.
`SEGMENT`	No	String (100)	Catalog segment or group to which the article belongs to. Example: Plumbing supplies, Electrical supplies In SAP Ariba solutions, you can map this field value to an existing master data field by creating catalog content enrichment rules.

ARTICLE_FEATURES

SAP Ariba supports the ARTICLE_FEATURES element within the ARTICLE element and consists of data fields which classify an article.

Element/Attribute	Mandatory	Type and length	Notes

ARTICLE_FEATURES	No	-	Used to define the classification of the article.
			`<ARTICLE>`
			`.`
			`<ARTICLE_FEATURES>`
			`<REFERENCE_FEATURE_SYSTEM_NAME>...</REFERENCE_FEATURE_SYSTEM_NAME>`
			`<REFERENCE_FEATURE_GROUP_ID>...</REFERENCE_FEATURE_GROUP_ID>`
			`or`
			`<REFERENCE_FEATURE_GROUP_NAME>...<REFERENCE_FEATURE_GROUP_NAME>`
			`</ARTICLE_FEATURES>`
			`.`
			`</ARTICLE>`

REFERENCE_FEATURE_SYSTEM_NAME	No	String (50)	Specifies the classification or feature group system used to classify the article.
REFERENCE_FEATURE_GROUP_ID or REFERENCE_FEATURE_GROUP_NAME	No	String (60)	Specifies the classification ID or group associated with the specified classification system.

FEATURE

The FEATURE element within the ARTICLE_FEATURES element describes the additional features or specifications of an item. You can specify multiple features of an item.

The following table lists the elements within the FEATURE element:

Element	Mandatory	Type and length	Notes

FNAME	Yes	String (60)	Indicates the unique name of a feature of an item.
			Example:
			`<FEATURE>`
			`  <FNAME>Length</FNAME>`
			`  <FVALUE>32</FVALUE>`
			`  <FUNIT>cm</FUNIT>`
			`</FEATURE>`

FVALUE	Yes	String (60)	Indicates the value of the specified feature. If the value contains any of the following special characters, precede the character with a backslash (\) to avoid catalog validation errors: • / • { • } • ; • = • , • " For example, if the value is 7,4, specify it as 7\,4.
FUNIT	No	String (20)	Indicates the unit of measure of the specified feature.

SAP Ariba solutions process the FNAME and FVALUE values that you specify as key-value pairs and map them to the **Item Spec** catalog field by default. The **Additional Specifications** section in the **Item Details** pages in the catalog user interface displays the key-value pairs as a table. FUNIT values when specified and automatically appended to the FVALUE value. If you specify multiple FVALUE values, they are displayed in the table seperated by commas.

The following lists a BMEcat catalog that describes the various specifications of an item using the FEATURE element:

```
<ARTICLE_FEATURES>
    <REFERENCE_FEATURE_SYSTEM_NAME>eclass-3.0</REFERENCE_FEATURE_SYSTEM_NAME>
    <REFERENCE_FEATURE_GROUP_ID>24-11-03-21</REFERENCE_FEATURE_GROUP_ID>
    <FEATURE>
        <FNAME>Width</FNAME>
        <FVALUE>15</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
    <FEATURE>
        <FNAME>Height</FNAME>
        <FVALUE>8</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
    <FEATURE>
        <FNAME>Length</FNAME>
        <FVALUE>32</FVALUE>
        <FUNIT>cm</FUNIT>
    </FEATURE>
    <FEATURE>
        <FNAME>Colors</FNAME>
```

```
        <FVALUE>red</FVALUE>
        <FVALUE>blue</FVALUE>
        <FVALUE>green</FVALUE>
        <FVALUE>yellow</FVALUE>

    </FEATURE>
</ARTICLE_FEATURES>
```

> **Note**
>
> - The `VARIANTS` feature within the `FEATURE`element is not supported.
> - When specifying feature names in the `FNAME` element, names that are defined in `FT_NAME` under the `FEATURE_TEMPLATE` element are not supported.

`ARTICLE_ORDER_DETAILS`

SAP Ariba supports `ARTICLE_ORDER_DETAILS` in `ARTICLE` in `T_NEW_CATALOG`.

Element/Attribute	Mandatory	Type and length	Notes

`ARTICLE_ORDER_DETAILS`	Yes		Contains elements which define the terms and conditions that are applied when a particular article is ordered.
			`<ARTICLE>`
			`  .`
			`  <ARTICLE_ORDER_DETAILS>`
			`  <ORDER_UNIT>...</ORDER_UNIT>`
			`  <PRICE_QUANTITY>...</PRICE_QUANTITY>`
			`  <QUANTITY_MIN>...</QUANTITY_MIN>`
			`  <QUANTITY_INTERVAL>...</QUANTITY_INTERVAL>`
			`  </ARTICLE_ORDER_DETAILS>`
			`.`
			`</ARTICLE>`

`ORDER_UNIT`	Yes	Unit (3)	Unit in which the article can be ordered.
`PRICE_QUANTITY`	No	Number	The quantity corresponding to the unit price of the item.
`QUANTITY_MIN`	No	Integer	Indicates the minimum order quantity of an item
`QUANTITY_INTERVAL`	No	Integer	Indicates the quantity interval for ordering the item.

ARTICLE_PRICE_DETAILS

SAP Ariba supports `ARTICLE_PRICE_DETAILS` in `ARTICLE` in `T_NEW_CATALOG`.

Element/Attribute	Mandatory	Type and length	Notes
`ARTICLE_PRICE_DETAILS`	Yes		Specifies price-related data. `<ARTICLE_PRICE_DETAILS>   <DATETIME type="valid_start_date">   <DATE>...</DATE>   </DATETIME>   <DATETIME type="valid_end_date">   <DATE>...</DATE>   </DATETIME>   <ARTICLE_PRICE price_type="net_customer">   .   </ARTICLE_PRICE> </ARTICLE_PRICE_DETAILS>`
`DATETIME type = "valid_start_date"`	No	datetime	SAP Ariba Procurement solutions validate that the end date is not before the start date. Catalogs that fail this validation are not uploaded.
`DATETIME type = "valid_end_date"`	No	datetime	Date when the validity of the price ends.
`ARTICLE_PRICE price_type="net_customer"`	Yes	String (20)	SAP Ariba only supports the `net_customer` value for price type. Any other value is ignored.

ARTICLE_PRICE

SAP Ariba supports ARTICLE_PRICE in ARTICLE_PRICE_DETAILS.

Element/Attribute	Mandatory	Type and length	Notes

ARTICLE_PRICE price_type="net_customer"	Yes	String (20)	SAP Ariba only supports the net_customer value for price type. Any other value is ignored.
			
			`<ARTICLE_PRICE_DETAILS>`
			`  .`
			`  <ARTICLE_PRICE price_type="net_customer">`
			`  <PRICE_AMOUNT>...</PRICE_AMOUNT>`
			`  <PRICE_CURRENCY>...</PRICE_CURRENCY>`
			`  <PRICE_FACTOR>...</PRICE_FACTOR>`
			`  <LOWER_BOUND>...</LOWER_BOUND>`
			`  <TAX>...</TAX>`
			`  </ARTICLE_PRICE>`
			`  .`
			`</ARTICLE_PRICE_DETAILS>`

PRICE_AMOUNT	Yes	Number	Indicates the amount of the price.
PRICE_CURRENCY	No	String (3)	Indicates the currency of the price.
PRICE_FACTOR	No	Number	The (discount) factor that is multiplied by the priceamount value to determine the end price.
LOWER_BOUND	No	Number	The lower limit of the item quantity for which the price is applied.
TAX	No	Number	A factor for sales tax applicable to the item price. Example: A value of 0.16 indicates 16 percent. **Note** This field is supported in SAP Ariba Catalog only.

MIME_INFO

SAP Ariba supports the `MIME` element in `MIME_INFO` in both `T_NEW_CATALOG` and `T_UPDATE_PRODUCTS`, with the following child elements:

Element/Attribute	Mandatory	Type and length	Notes

`MIME_INFO`	No	-	References to additional multimedia documents on an article can be specified.
			`<ARTICLE>`
			`.`
			`<MIME_INFO>`
			`<MIME>`
			`<MIME_TYPE>...</MIME_TYPE>`
			`<MIME_SOURCE>...</MIME_SOURCE>`
			`<MIME_DESCR>...</MIME_DESCR>`
			`<MIME_ALT>...</MIME_ALT>`
			`<MIME_PURPOSE>...</MIME_PURPOSE>`
			`</MIME>`
			`</MIME_INFO>`
			`.`
			`</ARTICLE>`

`MIME`	Yes	-	The MIME element serves to transfer information about one multimedia file. The file itself is only referenced and must be transferred separately.
`MIME_TYPE`	No	String (30)	SAP Ariba supports all of the attributes for `MIME_TYPE`. SAP Ariba Procurement solutions display the following `MIME_TYPE`s in the catalog item details **Related Documents** area: • `application/pdf` • `url` • `text/html` • `text/plain` SAP Ariba Procurement solutions use the following `MIME_TYPE`s for thumbnail, detail, and full-size images: • `image/jpeg` • `image/gif`
`MIME_SOURCE`	Yes	String (250)	Indicates the source of the document. The value can be a URL or the pathname to a document.
`MIME_DESCR`	No	String (250)	Indicates the description of the document.
`MIME_ALT`	No	String (50)	Indicates the alternative text for the document.

Element/Attribute	Mandatory	Type and length	Notes
`MIME_PURPOSE`	No	String (20)	Although the BMEcat standard does not denote `MIME_PURPOSE` as a mandatory element, it is mandatory for catalogs uploaded to SAP Ariba Procurement solutions, where it is used to determine where to display the content. SAP Ariba supports the following `MIME_PURPOSE` attributes for images: • `thumbnail` • `normal` • `detail` **Note** If you specify a thumbnail image for an item, you must also specify an associated normal or detail image. SAP Ariba Procurement solutions do not allow catalog items that only have a thumbnail image. SAP Ariba supports the following `MIME_PURPOSE` attribute for related documents: • `data_sheet`
`MIME_ORDER`	Yes	Integer	Indicates the order in which the documents are to be displayed. Although the BMEcat standard does not denote `MIME_ORDER` as a mandatory element, it is mandatory for catalogs uploaded to SAP Ariba Procurement solutions, where it is used to determine how to display multiple images or related documents for an item. If the same image is used for multiple image types, they should all have the same `MIME_ORDER`.

ARTICLE_REFERENCE

SAP Ariba supports the `ARTICLE_REFERENCE` element in `T_NEW_CATALOG`.

Element/Attribute	Mandatory	Type and length	Notes

`ARTICLE_REFERENCE type="..."`	No	String (20)	Used to refer from one article to another.
			SAP Ariba supports the following values for `ARTICLE_REFERENCE type`:
			• `mandatory`
			• `similar`
			• `sparepart`
			• `accessories`
			• `followup`
			`<ARTICLE>`
			`.`
			`<ARTICLE_REFERENCE type="...">`
			`<ART_ID_TO>...</ART_ID_TO>`
			`</ARTICLE_REFERENCE>`
			`.`
			`</ARTICLE>`

`ART_ID_TO`	Yes	String (32)	SAP Ariba Procurement solutions only use `ART_ID_TO` to identify related items. All related items must be in the same catalog. SAP Ariba Procurement solutions do not support related items from different catalogs.

Customer Catalogs in Microsoft Excel (.XLSX) Format

SAP Ariba solution sites configured for managing customer catalogs using the content management system (CMS) support customer catalog files in Microsoft Excel (`.xlsx`) format.

Excel catalogs are created as Microsoft Excel workbooks and must be saved with a `.xlsx` extension. Every Excel workbook file has two sheets, a Headers sheet, and an Items sheet. The Headers sheet specifies general attributes that apply to all the data in the file and the Items sheet describes the items in the customer catalog.

You can use any standard ASCII characters in the customer catalog filename. Use the extension `.xslx` for Excel catalogs.

> **Note**
>
> SAP Ariba solutions support Excel (`.xlsx`) catalog files only in sites configured for managing customer catalogs using the content management system (CMS).

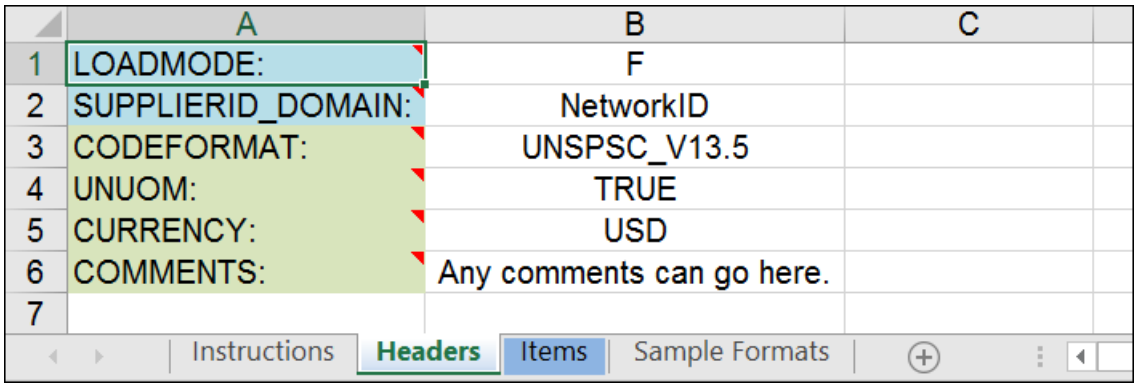
Excel Headers sheet

The Headers sheet specifies general attributes that apply to all the data in the file.

The following table lists the keys recognized in the Excel catalog file header sheet:

Key Name	Type	Description
`LOADMODE:`	Required	Indicates whether to load the entire file or reload just the items that are new or have been changed. Loading all items is called full loading; loading only new or changed items is called incremental loading. Valid values are `F` (for Full), and `I` (for Incremental). Default value is `F`.
`SUPPLIERID_DOMAIN:`	Required	Indicates the identification system you use to identify suppliers in your catalogs. You can use any identification system, such as D-U-N-S, **Ariba Network ID**, or customer-specific Supplier IDs. Example: `NetworkID`

Key Name	Type	Description
`CODEFORMAT:`	Required	Specifies the default commodity code system. Example: **UNSPSC**
`UNUOM:`	Required	Indicates whether this file uses United Nations Units of Measure (UNUOM). If unspecified or set to FALSE, the file is assumed to use American National Standards Institute (ANSI) units of measure. SAP Ariba recommends you set this value to **TRUE** to use UNUOM.
`CURRENCY:`	Required	Specifies the currency for interpreting prices in the current file. Must be an ISO 4217 currency name. Example: **USD**
`COMMENTS:`	Optional	Indicates lines of plain text.



Excel catalog Items sheet

The Items sheet describes the items in the customer catalog. Each row in the sheet indicates a catalog item.

The following table lists the keys recognized in the Excel catalog file Items sheet:

Key Name	Key Name Attribute	Required/Optional	Description
`Supplier ID`		Required	Indicates the supplier ID. Typically a D-U-N-S or NetworkID number.

Key Name	Key Name Attribute	Required/Optional	Description
`Supplier Part ID`		Required	Indicates the supplier part ID
`Manufacturer Part ID`		Optional	Indicates the manufacturer part ID
`Item Description`		Required	The description of the customer catalog item.
`Unit Price`		Required	Indicates the cost of the item.
`Unit of Measure`		Required	specifies how the item is packaged or measured;
`Lead Time`		Optional	Number of business days between the time the order is received and the time it is delivered to the customer.
`Short Name`		Required	Short title for the item, to be used in lists where `Item Description` is too long.
`Manufacturer Name`		Optional	Name of the manufacturer.
`Supplier URL`		Optional	Web address of the supplier's data sheet.
`Manufacturer URL`		Optional	Web address of the manufacturer's data sheet.
`Market Price`		Optional	List price or suggested retail price. This field has the same requirements as `Unit Price`.
`Classification Codes-1`	`Domain` `Value`	Optional	Domain-value pairs.
`Classification Codes-2`	`Domain` `Value`	Optional	Domain value pairs.
`Supplier Part Auxiliary ID`		Optional	Part of the unique ID for an item, when combined with `Supplier ID` and `Supplier Part ID`.

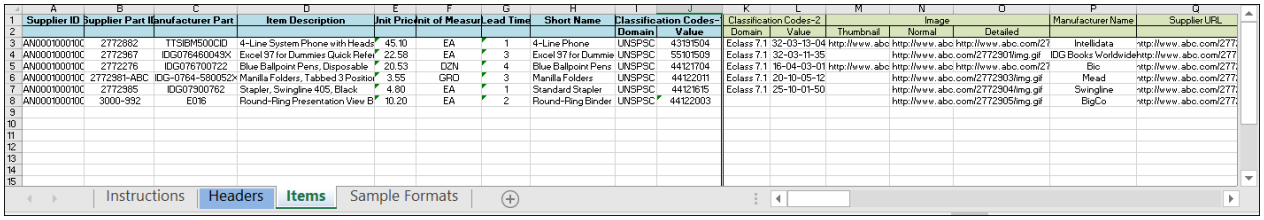
Key Name	Key Name Attribute	Required/Optional	Description
`Language`		Optional	Java locale code for the language used in the `Item Description`. Example: `en-US`
`Currency`		Optional	The currency of the item. This value overrides `Currency` in the header.
`Image`	`Detailed`	Optional	URL or pathname to a detailed image, Images must be in JPEG, GIF, PNG, or BMP format.
	`Thumbnail`		URL or pathname to a thumbnail image, Images must be in JPEG, GIF, PNG, or BMP format.
	`Normal`		URL or pathname to a full-size image. Images must be in JPEG, GIF, PNG, or BMP format. For more information, see Thumbnail, Detail, and Full-Size Images in Excel Catalogs [page 75].
`Expiration Date`		Optional	Date that the item is no longer valid, in YYYY-MM-DD format. This date must be in the future.
`Effective Date`		Optional	Date that the item becomes valid, in YYYY-MM-DD format. This date must be prior to the expiration date.
`PunchOut Enabled`		Optional	Specifies a PunchOut index catalog item. If true, `Supplier URL` becomes the URL of the PunchOut catalog. The default value is `False`.

Key Name	Key Name Attribute	Required/Optional	Description
PunchoutLevel		Optional	Determines how the procurement application displays the PunchOut index catalog item. Can be store, aisle, shelf, or product.
Territory Available		Optional	List of one or more ISO country or region codes. SAP Ariba recommends that you use ISO 3166-1 and 3166-2 codes for this field.
Delete		Optional	Key that indicates that this item is to be deleted when used an incremental catalog.
IsPartial		Optional	Set to yes to indicate that the item can be partially specified in purchase orders.
isInternalPartID		Optional	Set to yes to indicate that the catalog item is a non-standard item.
DefaultRelevance		Optional	Partially determines the relevance ranking the procurement application uses to display catalog items in a search. For example, 1000 is most relevant, and 0 is least relevant.
Price Unit Quantity		Optional	The quantity corresponding to the unit price of the item specified by the supplier. The unit price is based on the price unit quantity of the item.
Price Unit		Optional	The unit of measure for the unit price.
Unit Conversion		Optional	Ratio used to convert the ordered unit to the price unit. The unit price is based on the price unit quantity and the price unit.

Key Name	Key Name Attribute	Required/Optional	Description
`Price Unit Quantity Description`		Optional	Any information for advanced pricing details. Use this field to store additional information on the unit conversion calculation.
`InKitOnly`		Optional	
`Promotion Rank`		Optional	
`IsPreferredItem`		Optional	Set to `yes` to indicate that this item is from a preferred supplier.
`hazardousmaterials`		Optional	Set to `yes` to indicate that this item is classified as a hazardous material.
`bestprice`		Optional	
`green`		Optional	Set to `yes` to indicate that this item is classified as a green item.
`PriceConfiguration-1`	`StartDate`	Optional	You can define different levels of pricing for items based on different quantities and dates(validity period). Every different combination of these values can also define a different price. `StartDate` indicates the start date of the validity period.
	`EndDate`		Indicates the end date of the validity period.
	`PriceType`		Indicates the type of the price. SAP Ariba solutions support `net_customer` as the value that indicates that the price is customer-specific and excludes sales taxes.
	`Amount`		The amount of the price.

Key Name	Key Name Attribute	Required/Optional	Description
	PriceCurrency		The currency of the price.
	PriceFactor		The (discount) factor that is multiplied by the priceamount value to determine the end price.
	Tax		The tax rate applicable to the item price. A value of 5 indicates 5 percent.
	Lowerbound		The lower limit of the item quantity for which the price is applied.
PriceConfiguration-2		Optional	See, PriceConfiguration-1
PriceConfiguration-3		Optional	See, PriceConfiguration-1
Attachments-1	Source	Optional	Indicates the source of the document. The value can be a URL or the pathname to a document.
	Descr		Indicates the description of the document.
Attachments-2			See, Attachments-1
RelatedItems-1	Supplier Part ID	Optional	Indicates the source of the document. The value can be a URL or the pathname to a document.

Key Name	Key Name Attribute	Required/Optional	Description
	`Type`		Indicates the type of the item in relation to the main item. Ariba Procurement and Invoicing Solutions support the following values for related items: • `mandatory` • `similar` • `sparepart` • `accessories` • `followup`
`RelatedItems-1`			See, `RelatedItems-1`
`RelatedItems-2`			See, `RelatedItems-1`
`Parametric Name`			Textual label for the type of data you supplied in the Parametric Data field. Required if you specify Parametric Data.
`Parametric Data`			Key-Value pairs of attributes of the parametric type. For example: `{Color=Red; Size=44;}`



Customer Catalog Internationalization and Localization

Customer Catalog Language [page 147]

Creating Multilingual CIF and cXML Catalogs [page 148]

Customer Catalog Character Set and Character Encoding [page 153]

Customer Catalog Language

Internationalization and localization involve specifying the language or languages you use to describe customer catalog items. You specify customer catalog language using Java locale codes (CIF and cXML) or ISO 639-2 language codes (BMEcat).

You can describe customer catalog items in any language. Both CIF and cXML allow you to specify the language through Java locale codes, such as `fr_FR` for French France and `en_US` for English US. BMEcat catalogs allow you to specify the language through ISO 639-2:1998 language codes, such as `eng` for English and `deu` for German. If you do not specify a Java locale, the default locale set by the buying organization is used.

In CIF and cXML catalogs, you must include the country portion of the locale code for display names used with types, attributes, and enumerations in type definitions. For example, if the user's locale is `fr_CA` (French Canadian) and an enumeration value contains only an entry for `fr_FR` (French France), SAP Ariba Procurement solutions do not default to the `fr_FR` display name, but will default to the locale set by the buying organization. Therefore, if you want specific country-language granularity for display names, you must explicitly include each country-language locale code in the type definition. To learn more about type definitions see, Type Definition File Format [page 158].

The following example shows a cXML example of a enumeration value with the locale code included:

```
<EnumerationValue>
    <Extrinsic name="CanonicalValue">half coupling</Extrinsic>
    <Name xml:lang="en-US">Half Coupling</Name>
    <Name xml:lang="fr-FR">Demi Accouplement</Name>
</EnumerationValue>
```

Locale codes

For the names of languages in CIF and cXML catalogs, use Java locale codes, for example, `it_IT`, `fr_FR`, and `en_US`. For the names of languages in BMEcat catalogs, use ISO 639-2 language codes, for example, `deu`, `fre`, and `eng`.

For a list of Java codes, see:

```
http://java.sun.com/j2se/1.4.2/docs/guide/intl/locale.doc.html
```

CIF uses an underscore (_) within the codes; for example `it_IT`, `fr_FR`, and `en_US`. 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.

For a list of ISO 639-2 language codes, see:

```
http://www.loc.gov/standards/iso639-2/
```

ISO language codes do not specify different regional variations of a language.

Related Information

Creating Multilingual CIF and cXML Catalogs [page 148]
Customer Catalog Character Set and Character Encoding [page 153]

Creating Multilingual CIF and cXML Catalogs

A multilingual customer catalog contains descriptions in two or more languages. There are several ways to create multilingual CIF and cXML catalogs based on the viewing and searching preferences of your customers, and what information they want carried forward to requisitions and purchases.

In all the methods, 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 Customer Catalogs

A buying organization might want you to deliver multilingual customer catalogs, for example, containing descriptions in both French and English. For every SKU item, create two customer 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 Aux 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 the following example, which contains `UOM` and `Language`.) For more information about the Supplier Part Auxiliary ID field, see Supplier Part Auxiliary IDs [page 18].
- **Language**: used by SAP Ariba Procurement solutions to filter for the English or French description products. For example, enter `en_EN` for the English description line item.

To include prices in the local currency, use the `Currency` field. See Sample CIF file segment with Supplier Part Aux ID and Language and Currency Fields for an example of specifying local currency.

In addition to specifying the language, the buying organization must add a custom filter to their procurement solution from SAP Ariba so that the appropriate language displays. The custom filter must match the user's locale and the entry in the `Language` field.

Sample CIF file segment with Supplier Part Aux ID and Language and Currency fields

The sample file segment illustrates the use of the `Supplier Part Aux ID` and language and currency specifications.

Supplier ID	Supplier Part ID	Item Description	Product Code	Unit Price	Unit of Measure	Supplier Part Aux ID	Language	Currency
0001	01	Blue Pen	12133 76	1.9 5	EA	EA-en_GB	en_GB	GBP
0001	01	Blue Pen	12133 76	3.0 0	DOZ	DOZ-en_GB	en_GB	GBP
0001	01	Stylo Bleu	12133 76	1.2 0	EA	EA-fr_FR	fr_FR	FRF
0001	01	Stylo Bleu	12133 76	2.7 5	DOZ	DOZ-fr_FR	fr_FR	FRF

Advantages and disadvantages of multilingual catalogs

- Search advantage: Users can search using their local language, and the search engine in SAP Ariba Procurement solutions can use additional logic to find logical misspellings or phonetic matches.
- Display potential disadvantage: Only the local language description displays.
- Carrying through to requisition, potential disadvantage: Only the local language description appears on requisitions and purchase orders.

Related Information

Multilingual Customer Catalogs with Territory Available Field [page 150]
Multilingual Customer Catalogs Combining Languages on One Line [page 151]

Multilingual Customer Catalogs with Territory Available Field

The `Territory Available` field enables you to create multilingual customer catalogs that can be searched in different languages by customers in a single region or country.

Customers 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 France office. English users must be able to search in English but order items using the Euro pricing.

To handle this situation, create two catalog items for every SKU item, 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 following example contains both `UOM` and `Language`.) For more information about the Supplier Part Auxiliary ID field, see Supplier Part Auxiliary IDs [page 18].
- **Language**: Used by SAP Ariba Procurement solutions 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 [page 43].

Use the `Currency` field to include prices in the local currency. In the following example, the price is in the local currency.

The buying organization must also add custom filters to their procurement solution from SAP Ariba to display the appropriate language and allow users to search only for items available in their territory. The custom filters must match users' locales for both the `Language` and `Territory Available` fields.

Sample CIF file segment illustrating Territory Available field

The sample file segment illustrates the use of the `Supplier Part Aux ID`, `Language`, `Territory Available`, and `Currency` specifications.

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

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	1213 376	2. 75	DOZ	DOZ- en_GB	en_GB	FR	FRF
0001	01	Stylo Bleu	1213 376	1. 20	EA	EA- fr_FR	fr_FR	FR	FRF
0001	01	Stylo Bleu	1213 376	2. 75	DOZ	DOZ- fr_FR	fr_FR	FR	FRF

Advantages and disadvantages of multilingual catalogs

- Search advantage: Users can search using their local language and the search engine in their procurement solution from SAP Ariba can use additional logic to find logical misspellings or phonetic matches.
- Display potential disadvantage: Only the local language description displays.
- Carrying through to requisition, potential disadvantage: Only the local language description appears on requisitions and purchase orders.

Related Information

Multilingual Customer Catalogs with Territory Available Field [page 150]
Multilingual Customer Catalogs Combining Languages on One Line [page 151]
Customer Catalog Character Set and Character Encoding [page 153]

Multilingual Customer Catalogs Combining Languages on One Line

A buying organization might want you to supply customer 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.

Specifying multiple languages in one line using the Manufacturer URL field

You can enter the English description in the `Item Description` field and enter the French description in the `Manufacturer URL`. The `Manufacturer URL` field is recommended because it can contain up to 255 characters.

> **Note**
>
> You can enter the second language description in unused fields other than `Manufacturer URL` if necessary.

The sample file segment illustrates the use of the `Manufacturer URL` to specify dual display languages.

Table 2: Sample CIF file segment using `Manufacturer URL`

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	12133 76	1. 95	EA	EA	Stylo Bleu	GBP
0001	01	Blue Pen	12133 76	3. 00	DOZ	DOZ	Stylo Bleu	GBP

Advantages and disadvantages of using Manufacturer ID for multi-language display

- Searching: Users can search on both English and French descriptions. An SAP Ariba solution must be customized to search against the `Manufacturer URL` field. The search engine in an SAP Ariba solution can use additional logic to find logical misspellings or phonetic matches only for the `Item Description` field.
- Displaying: Both languages display.
- Carrying through to requisition: an SAP Ariba solution must be configured to include `Manufacturer URL` as an extrinsic tag on requisitions and purchase orders because it does not include that field by default.

Specifying multiple languages in one line using the Description field

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

The sample file segment illustrates the use of the `Item Description` to display dual languages.

Table 3: Sample CIF file segment with dual languages in Item Descriptions

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	121337 6	1.95	EA	EA	GBP
0001	01	Blue Pen / Stylo Bleu	121337 6	3.00	DOZ	DOZ	GBP

Advantages and disadvantages of using Item Description to display dual languages

- Searching: Users can search on both English and French descriptions, without additional customization of their procurement solution from SAP Ariba. The search engines in SAP Ariba Procurement solutions cannot use additional logic to find logical misspellings or phonetic matches.
- Display: Both languages display.
- Carrying through to requisition: Both language descriptions can appear on requisitions and purchase orders. The `Description` field accommodates 2000 characters (2000 bytes in SAP Ariba Procurement solutions), but only the first 255 characters appear on requisitions and purchase orders.

Related Information

Creating Multilingual CIF and cXML Catalogs [page 148]

Customer Catalog Character Set and Character Encoding

When creating customer catalogs, understand the implications of the character set and character encoding you use. Different versions of SAP Ariba Procurement solutions support different character sets. If the correct character set and encoding is not specified, your catalog might not appear as you expect.

Customer Catalog character set

You can specify the character set for CIF, cXML, or BMEcat catalog uses. If you do not specify a character set, SAP Ariba solutions use the `Cp1252` (Windows Latin-1) character set to display customer catalog contents.

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

The following table notes the character sets supported in customer catalogs by SAP Ariba solutions and SAP Business Network:

Description	Code
Western European languages	`8859_1`
	`Cp1252`
Japanese	`Shift_JIS`
	`MS932`

Description	Code
Simplified Chinese	GB2312
	EUC_CN
	MS936
Traditional Chinese	Big5
	EUC_TW
	MS950
Korean	KSC5601
	EUC_KR
	MS950

- The code UTF-8 includes all the character sets.
- Suppliers providing Japanese content must use Zenkaku Katakana characters, not Hankaku Katakana characters. Hankaku Katakana uses more space in procurement application UTF-8 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 solutions from SAP Ariba, UTF-8 non-ASCII characters expand to a number of bytes larger than the number of original characters. Extended-ASCII characters generally expand to two bytes, and Asian characters to three bytes.

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

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 CHARSET value or cXML encoding attribute must be set to UTF-8.

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

> **Note**
>
> 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 email your catalogs, make sure that the encoding of the email matches the encoding of the catalog, such as `UTF-8`.

> **Note**
>
> `Shift_JIS` encoding does not include some extended Japanese characters. To make the greatest number of types of characters available, use `UTF-8` or `MS932` encoding.

Type Definition Files

Overview of Type Definition Files [page 156]

Type Definition File Format [page 158]

Primitive Types [page 161]

Parametric Data Examples [page 169]

Supplemental Attribute Examples [page 172]

Overview of Type Definition Files

SAP Ariba solutions use cXML files called type definition files to define parametric search attributes and supplemental customer catalog attributes.

Buying organizations can use type definition files to extend the customer catalog format and provide additional custom functionality. Type definition files extend the customer catalog format used by SAP Ariba solutions. They define types, elements used to create custom catalog fields with specific attributes.

A `Type` is the basic element within type definition files which specifies the name of the type and the object it extends. By using types defined in type definition files, buying organizations can provide custom catalog fields that allow suppliers to add parametric data or supplemental attributes to their catalogs.

SAP Ariba solutions use the following type definition files:

- Parametric data types
- Supplemental attribute types
- Catalog kitting types
- Procurement category types

These files are required by SAP Ariba solutions regardless of whether parametric data or supplemental attribute data comes from CIF or cXML catalogs.

Type definitions and attributes are managed and maintained by the SAP Ariba Catalog Experts during the deployment of your SAP Ariba solution or later through the SAP Ariba Customer Support.

For an example of a type definition file, see Basic Type Definition Format [page 158].

This section provides an overview of parametric data types and supplemental attributes in SAP Ariba Buying solutions.

> **Note**
>
> By default, SAP Ariba solutions do not recognize and use parametric data or supplemental attributes provided in customer catalogs. Your site must be customized by the SAP Ariba Enablement team or by SAP Ariba Customer Support.

Parametric Data Types

Parametric data types define the parametric names and attributes for commodity-specific search data.

The following CIF example shows a customer catalog item with three parametric attributes:

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

`TYPE`, `MATERIAL`, and `THICKNESS` are data types. They define the type of search data, but not the data itself. The search data is part of the individual line items in catalogs. For an overview of parametric data, see Parametric Search Data [page 48].

> **Note**
>
> An SAP Ariba solution ignores `SearchGroup` elements in cXML catalogs.

For more information about importing and managing parametric types in an SAP Ariba solution, see the *Procurement and Invoicing Service Administration Guide*.

Supplemental Attribute Types

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.

Parametric and Supplemental Attribute Naming Restrictions

Parametric and supplemental attribute names have the regular restrictions.

For more information, see Parametric Attribute Names and Data Types [page 51] and Supplemental Attributes [page 47].

In addition, an SAP Ariba solution 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
IsPartial
```

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

Catalog Kitting Types

Your SAP Ariba solution also contains a type used by catalog kits. This type definition file appears as the `SystemKittingType` on the type definition screens in Ariba Administrator. Any customization to this type must be made only under the guidance of an SAP Ariba representative.

For more information about catalog kits, see the catalog administration guide for your SAP Ariba solution.

Related Information

Type Definition File Format [page 158]
Primitive Types [page 161]

Type Definition File Format

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

Basic Type Definition Format

`TypeDefinition` documents contain two elements, `TypeProvider` and `Type`.

`TypeProvider` identifies the organization creating the types. `Type` specifies the names and other attributes of custom customer catalog fields.

The syntax of `TypeDefinition` documents is described by the cXML Document Type Definition (DTD) `Catalog.dtd`, available at 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 version="1.0" encoding="UTF-8"?>
<!DOCTYPE TypeDefinition SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/
Catalog.dtd">
<TypeDefinition>
```

```
    <TypeProvider name="System">
        <OrganizationID>
            <Credential domain="buyersystemid">
                <Identity>1</Identity>
            </Credential>
        </OrganizationID>
    </TypeProvider>
    <Type name="CUPS" extends="System:CatalogItem">
        <Name xml:lang="en-US">Cups</Name>
        <Name xml:lang="de-DE">Becher</Name>
        <Description xml:lang="en-US">Standard Drinking Cups</Description>
        <Description xml:lang="de-DE">Standard Becher</Description>
        <TypeAttribute name="MATERIAL" type="Name" isRefinable="yes">
            <Name xml:lang="en-US">Material</Name>
            <Name xml:lang="de-DE">Material</Name>
        </TypeAttribute>
        <TypeAttribute name="OUNCES" type="Number" isRefinable="yes">
            <Name xml:lang="en-US">Ounces</Name>
            <Name xml:lang="de-DE">Unzen</Name>
        </TypeAttribute>
    </Type>
</TypeDefinition>
```

The major elements in this file are described in the following sections.

DOCTYPE

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

TypeProvider

The cXML standard allows any value in the `TypeProvider name` attribute, but the SAP Ariba solution accepts only the name `System` for this attribute.

The SAP Ariba solution does not check the type provider's credential, which is only used to identify the provider.

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 Attribute Names and Data Types [page 51]. `"CatalogItem"` for supplemental attributes.
`extends` (required)	Specifies the SAP Ariba solution object to which to apply the type: `"System:CatalogItem"` for parametric data. `"System:CatalogItemInternal"` for supplemental attributes.

`Type` contains the following elements:

Element	Description
`Description`	Specifies a description for the type. Create a useful description that differentiates this type from others. This value is visible to catalog managers and the SAP Ariba solution's administrators, not end users. For multilingual configurations, you can specify a name for each locale. Recommended maximum length is 1024 characters.
`Name`	Specifies the user-visible name for this attribute. For multilingual configurations, you can specify a name for each locale.
`TypeAttribute`	Specifies the attributes for this data type. For a description of the properties you can use with this tag, see the following section.

Type attributes

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

`TypeAttribute` has the following properties:

Property	Description
`name`(required)	Specifies the canonical name of the attribute. For more information, see Parametric Attribute Names and Data Types [page 51]. For naming restrictions, see Parametric and supplemental attribute naming restrictions [page 157].
`type`(required)	Defines the data type for the attribute.
`isCaseSensitive`	Set to `yes` to preserve letter case 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 examples of syntax for collections such as enumeration values, see Enumerated Values in TypeAttribute [page 163].
`isHidden`	Set to `yes` to prevent the attribute from appearing for users. Use `isHidden` when you want to use an attribute for custom filters but do not want it to appear 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.
`isReadOnly`	Set to `yes` to prevent the attribute from being changed by the SAP Ariba solution's catalog managers or administrators.
`isRefinable`	Set to `yes` to display the attribute value in a pull-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 this attribute. If you set this property to `yes`, each catalog is re validated and only reactivated if the catalog items do not have empty values for this attribute. However, if the `IsPartial` flag is set to `true` in the catalog, empty `isRequired` values are not flagged during initial validation, but must be filled out when the item is ordered.

Property	Description
`isRequiredForOrdering`	Set to `yes` to indicate that this attribute must be specified to order the item. Any value provided for this attribute appears as an editable default value when a user orders the item. Set this property to require a value for this field in the order information sent to the supplier.
`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 and not available for search.
`mappedFrom`	(Supplemental attributes only) If this attribute is defined implicitly from an internal SAP Ariba solution object, use this property to specify the location of that object. You can map any attribute from the `CommonSupplier` and `Subscription` objects in an SAP Ariba solution. For example, if you want to add the common supplier's business type to the catalog item, you would pull that value from the common supplier object instead of adding the data to the catalog itself: `<TypeAttribute name="BusinessType" type="Id" isSearchable="yes" isRefinable="yes" mappedFrom="Supplier.BusinessType"> <Name xml:lang="en-US">Business Type</Name></TypeAttribute>` The value is loaded when the catalog is imported. Therefore, if the source value changes (such as the supplier name changes), you must reload the catalog or load a new version of the catalog to update the value in the catalog.
`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`.

Related Information

Primitive Types [page 161]
Parametric Data Examples [page 169]
Supplemental Attribute Examples [page 172]

Primitive Types

Primitive types describe basic data types, such as integers, strings, and Boolean values. They define the kind of data that is valid and can be stored for any given field.

A type is an object such as a shirt. It has attributes such as color and size. The kinds of values allowed for color and size are also types, but are referred to as primitive types. For example, color might be defined as a string while size might be defined as an integer. That means text values can be used for the color attribute while only integers (whole numbers) can be used for sizes.

Following are the primitive types you can specify for the attribute's `type` property.

Primitive Type	Description
`CurrencyCode`	A string containing between 1 and 5 characters.
`Date`	A date between 1970 and the present. **Note** A catalog item can be loaded with a custom date field. If a user selects the date while adding such an item to the cart, the date defaults to `00:00:00 GMT` during checkout. For example, if a user in Hawaii selects a date of `20/08/2023`, the date is reset to `20/08/2023 00:00:00 GMT`, which is `2:00 PM` on `19/08/2023` in Hawaii.
`Description`	A string containing between 1 and 2000 characters. 2000 characters is the recommended maximum limit for this type.
`Flag`	A Boolean containing between 1 and 5 characters.
`FutureDate`	A date in the future.
`Id`	A literal value containing between 1 and 128 characters. A literal value cannot be searched. Never use the characters `?`, `{`, and `}` in a field of this primitive type, as they cause validation errors.
`Integer`	A whole number, with no fraction.
`Name`	A string containing between 1 and 255 characters.
`Number`	A floating-point value.
`PartID`	A part number used for leading wildcarding.
`PositiveInteger`	An integer with a minimum value of 0 and a maximum value of 2147483647.
`Price`	A floating point value. Note that commas used as separators are not validated to make sure they are treated as thousand or decimal separators based on the file format.
`ShortName`	A string containing between 1 and 50 characters.
`TerritoryCode`	A literal value containing between 1 and 5 characters. A literal value cannot be searched. Never use the characters `?`, `{`, and `}` in a field of this primitive type, as they cause validation errors in catalog files.
`URL`	A string containing between 5 and 255 characters.

Related Information

Enumerated Values in TypeAttribute [page 163]
Canonical Values in TypeAttribute [page 168]
Range Values in TypeAttribute [page 168]

Enumerated Values in TypeAttribute

`TypeAttribute` elements can include a list of all possible values with `EnumerationValue` elements. Defining enumerations values validates a field against a specific set of values. For example, a pen might be available in only black, blue, and red.

Enumeration Value Example

The enumeration value example is part of a multilingual type definition file that uses the `EnumerationValue` element.

```
...
<TypeAttribute name="CouplingType" type="Name" isRefinable="yes" isSearchable="yes"
               isCollection="yes">
    <Name xml:lang="en-US">Coupling Type</Name>
    <
     EnumerationValue>

            <Extrinsic name="CanonicalValue">half coupling</Extrinsic>

            <Name xml:lang="en-US">Half Coupling</Name>

            <Name xml:lang="fr-FR">Demi Accouplement</Name>

        </EnumerationValue>

        <EnumerationValue>

            <Extrinsic name="CanonicalValue">full coupling</Extrinsic>

            <Name xml:lang="en-US">Full Coupling</Name>

            <Name xml:lang="fr-FR">Plein Accouplement</Name>

        </EnumerationValue>

        <EnumerationValue>
```

```
            <Extrinsic name="CanonicalValue">quick coupling</Extrinsic>

            <Name xml:lang="en-US">Quick Coupling</Name>

            <Name xml:lang="fr-FR">Accouplement Rapide</Name>

        </EnumerationValue>
</TypeAttribute>
```

> **Note**
>
> Notice the use of `CanonicalValue` to specify the canonical name of each enumeration value. The `CanonicalValue` of each enumeration value must be defined in the type definition file. The `CanonicalValue` element is not case sensitive.

Partial Item Enumeration Value Example

The `EnumerationValue` element can be used in conjunction with partial catalog items to display a dropdown list of allowable values when ordering an item.

The following example is a type definition file describing a partial item t-shirt that contains color, size, and material fields that must be ordered using specific allowable values.

```
         header information
         <TypeDefinition>
    <TypeProvider name="System">
        <OrganizationID>
            <Credential domain="buyersystemid">
                <Identity>1</Identity>
            </Credential>
        </OrganizationID>
    </TypeProvider>
    <Type name="TSHIRT" extends="System:CatalogItem">
        <Name xml:lang="en-US">T-shirt</Name>
        <Description xml:lang="en-US">Standard t-shirt</Description>
        <TypeAttribute name="COLOR" type="Name" isRefinable="yes"
isRequiredForOrdering="yes"
                   isSearchable="yes" isCollection="yes">
            <Name xml:lang="en-US">Color</Name>

                 <EnumerationValue>

                     <Extrinsic name="CanonicalValue">white</Extrinsic>

                     <Name xml:lang="en-US">White</Name>

                 </EnumerationValue>

                 <EnumerationValue>
```

```
                    <Extrinsic name="CanonicalValue">blue</Extrinsic>

                    <Name xml:lang="en-US">Blue</Name>
            </EnumerationValue>

                <EnumerationValue>

                    <Extrinsic name="CanonicalValue">red</Extrinsic>

                    <Name xml:lang="en-US">Red</Name>

                </EnumerationValue>

        </TypeAttribute>
        <TypeAttribute name="SIZE" type="Name" isRefinable="yes"
isRequiredForOrdering="yes"
                    isSearchable="yes" isCollection="yes">
            <Name xml:lang="en-US">Size</Name>

    <EnumerationValue>

                    <Extrinsic name="CanonicalValue">small</Extrinsic>

                    <Name xml:lang="en-US">Small</Name>

                </EnumerationValue>

                <EnumerationValue>

                    <Extrinsic name="CanonicalValue">medium</Extrinsic>

                    <Name xml:lang="en-US">Medium</Name>

                </EnumerationValue>

                <EnumerationValue>

                    <Extrinsic name="CanonicalValue">large</Extrinsic>

                    <Name xml:lang="en-US">Large</Name>

                </EnumerationValue>

            </TypeAttribute>
        <TypeAttribute name="MATERIAL" type="Name" isRefinable="yes"
isRequiredForOrdering="yes"
                    isSearchable="yes" isCollection="yes">
            <Name xml:lang="en-US">Material</Name>

                <EnumerationValue>
```

```
                    <Extrinsic name="CanonicalValue">cotton</Extrinsic>

                    <Name xml:lang="en-US">Cotton</Name>

                </EnumerationValue>

                <EnumerationValue>

                    <Extrinsic name="CanonicalValue">polyblend</Extrinsic>

                    <Name xml:lang="en-US">Polyblend</Name>

                </EnumerationValue>

        </TypeAttribute>
    </Type>
</TypeDefinition>
```

> **Note**
>
> If you do not set `isCollection` to `yes` on a type attribute with enumerations, only the first enumeration value appears on the item in the SAP Ariba catalog. Therefore, set `isCollection` to `yes` if you want users to see a list of all the enumerated values when ordering the item.

If a catalog file contains multiple items of the same type, include all possible type attributes and enumeration values in the type definition file and specify the attributes that apply to individual catalog items in the catalog file. For example, if you have a catalog containing three different t-shirt items with different colors and materials, specify the color `TypeAttribute` and material `TypeAttribute` of the `TSHIRT` type in the type definition file; list all possible values for each attribute and set `isRequiredForOrdering` to `yes`. Then in the catalog file, for each separate t-shirt item, specify the type attributes and enumeration values allowed for that particular t-shirt.

If a catalog file does not list any enumeration values for a `isRequiredForOrdering` item, but the type definition file does, the SAP Ariba solution lets the user choose from all values (up to 50 values maximum) listed in the type definition file when purchasing the item.

CIF Enumerative Values Example

The CIF example illustrates how to define enumerative values for a catalog.

```
CIF_I_V3.0

LOADMODE: F

CODEFORMAT: UNSPSC

SUPPLIERID_DOMAIN: buyersystemid

UNUOM: TRUE

CHARSET: UTF-8
```

```
FIELDNAMES: Supplier ID, Supplier Part ID, Manufacturer Part ID, Item Description, SPSC
Code, Unit Price, Unit of Measure, Lead Time, Language, Manufacturer Name, Supplier
URL, Manufacturer URL, Territory Avalible, Market Price, Parametric Name, Parametric
Data

DATA

7,S1,,Men’s Button Down Shirt,
53101602,15,ea,,en,,,,"(en_US,fr_FR)",,shirts,"{collared=true;color=(Red,Blue,Green);siz
e=(Large);}"

7,S2,,Men’s Tshirt,
53103000,25,ea,,en,,,,"(en_US,fr_FR)",,shirts,"{collared=false;color=(Blue,Green);size=(
Medium,Large);}"

7,S3,,Men’s Turtle Neck
Shirt,53101602,20,ea,,en,,,,"(en_US,fr_FR)",,shirts,"{collared=false;color=(Orange,Blue)
;size=(Small,Medium,Large);}"

ENDOFDATA
```

> **Note**
>
> When using collections for enumeration data, be sure to include double quotes around the parenthesized values. For example, "`(Value1,Value2,Value3)`"

cXML Enumerative Values Example

The cXML example is a partial catalog file, illustrating the declaration of enumerative values.

```
...
<IndexItemAdd>
    <ItemID>
        <SupplierPartID>MS3</SupplierPartID>
    </ItemID>
    <ItemDetail>
        <UnitPrice>
            <Money currency="USD">35</Money>
        </UnitPrice>
        <Description xml:lang="en-US">Men's Button Down Shirt</Description>
        <UnitOfMeasure>ea</UnitOfMeasure>
        <Classification domain="UNSPSC">53101602</Classification>
    </ItemDetail>
    <IndexItemDetail>
        <LeadTime>1</LeadTime>
        <SearchGroupData>
            <Name xml:lang="en-US">System:Shirts</Name>
            <SearchDataElement name="Color" value="red"></SearchDataElement>
            <SearchDataElement name="Color" value="blue"></SearchDataElement>
            <SearchDataElement name="Size" value="small"></SearchDataElement>
            <SearchDataElement name="Size" value="large"></SearchDataElement>
        </SearchGroupData>
    </IndexItemDetail>
</IndexItemAdd>
...
```

Canonical Values in TypeAttribute

A canonical value is a static, unique identifier that is independent of language or locale variations. Think of it as the system value, which can be different from the name or value displayed on the interface.

The `CanonicalValue` of each enumeration value must be defined in the type definition file and then language-specific values can be added for each locale. `CanonicalValue` name and value restrictions are the same as any other type attribute as described in Parametric Attribute Names and Data Types [page 51].

The localized value of the `CanonicalValue` is sent to suppliers in all purchase orders. If no localized value is provided for a particular locale, the canonical value is sent instead. Purchase orders sent in cXML contain the `CanonicalValue` value.

Range Values in TypeAttribute

`TypeAttribute` elements can include a range of possible values specified using the `Range` element.

For example, the following `Range` element defines the minimum and maximum number of ounces that can be specified.

```
...
<TypeAttribute name="OUNCES" type="Number" isRefinable="yes">
    <Name xml:lang="en-US">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.

> **Note**
>
> Your SAP Ariba solution does not support the `Range` element.

Parametric Data Examples

The CIF and cXML examples demonstrate how to define parametric data types for an SAP Ariba solution and how to use them in customer catalogs. Both require a type definition file separate from the actual customer catalog file.

> **Note**
>
> In a type definition file, do not use underscore (_) in the `Type name` field. If an underscore is used, the associated icons aren't displayed on the catalog search results page. For example, use `CUPTYPE` instead of `CUP_TYPE`.

Parametric Data Example Defining Two Attributes

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

Type definition files

```
<?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">
        <OrganizationID>
            <Credential domain="buyersystemid">
                <Identity>1</Identity>
            </Credential>
        </OrganizationID>
    </TypeProvider>
    <Comments> Parametric data fields for drinking cups </Comments>
    <Type name="CUPS" extends="System:CatalogItem">
        <Name xml:lang="en-US">Cups</Name>
        <Description xml:lang="en-US">Standard Drinking Cups</Description>
        <TypeAttribute name="MATERIAL" type="Name">
            <Name xml:lang="en-US">Material</Name>
        </TypeAttribute>
        <TypeAttribute name="OUNCES" type="Number">
            <Name xml:lang="en-US">Ounces</Name>
        </TypeAttribute>
    </Type>
</TypeDefinition>
```

CIF catalog example

```
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", 48101905,2.95,DZ,2,Dixie,,,
4.95,{MATERIAL="STYROFOAM";OUNCES=12;},CUPS
ENDOFDATA
```

cXML catalog example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Index SYSTEM "http://xml.cxml.org/schemas/cXML/1.2.008/cXML.dtd">
<Index loadmode="Full">
    <SupplierID domain="duns">771231234</SupplierID>
    <IndexItem>
        <IndexItemAdd>
            <ItemID><SupplierPartID>66056</SupplierPartID></ItemID>
            <ItemDetail>
                <UnitPrice><Money currency="USD">2.95</Money></UnitPrice>
                <Description xml:lang="en-US">
                    Large Hot/Cold Drinking Cups
                    </Description>
            <UnitOfMeasure>DZ</UnitOfMeasure>
                <Classification domain="UNSPSC">48101905</Classification>
                    <ManufacturerPartID>66056</ManufacturerPartID>
            <ManufacturerName>Dixie</ManufacturerName>
            </ItemDetail>
            <IndexItemDetail>
                <LeadTime>2</LeadTime>
                <SearchGroupData>
                    <Name xml:lang="en-US">CUPS</Name>
                    <SearchDataElement name="MATERIAL" value="STYROFOAM"/>
                    <SearchDataElement name="OUNCES" value=12/>
                </SearchGroupData>
            </IndexItemDetail>
        </IndexItemAdd>
    </IndexItem>
</Index>
```

Parametric Data Example Defining Three Attributes

This example shows parametric data named “Pipes” with three attributes named “Type”, “Length”, and “Material”.

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">
```

```
        <OrganizationID>
            <Credential domain="buyersystemid">
                <Identity>1</Identity>
            </Credential>
        </OrganizationID>
    </TypeProvider>
    <Type name="PIPES" extends="System:CatalogItem">
        <Name xml:lang="en-US">Pipes</Name>
        <Name xml:lang="fr-FR">Tuyaux</Name>
        <Description xml:lang="en-US">Pipes</Description>
        <TypeAttribute name="TYPE"  type="Name" isRefinable="yes">
            <Name xml:lang="en-US">Type</Name>
            <Name xml:lang="fr-FR">Espèce</Name>
        </TypeAttribute>
        <TypeAttribute name="SIZE" type="Number" isRefinable="yes">
            <Name xml:lang="en-US">Length</Name>
            <Name xml:lang="fr-FR">Longueur</Name>
        </TypeAttribute>
        <TypeAttribute name="WEIGHT_PER_FOOT" type="Number" isRefinable="yes">
            <Name xml:lang="en-US">Weight per Foot</Name>
            <Name xml:lang="fr-FR">Poids par Foot</Name>
        </TypeAttribute>
    </Type>
</TypeDefinition>
```

CIF catalog

```
CIF Header
DATA
599238,1234,1234,"Water Pipe",99998050,2.95,FT,2,Acme,,, 4.95,
{TYPE="SEAMLESS";SIZE=20\";"MATERIA"L= Steel;},PIPES
ENDOFDATA
```

cXML catalog

```
header information
<Index loadmode="Full">
    <IndexItem>
        <IndexItemAdd>
             . . .
            <ItemDetail>
                . . .
            </ItemDetail>
            <IndexItemDetail>
                . . .
                <SearchGroupData searchGroupName="Pipes">
                    <SearchDataElement name="TYPE" value="SEAMLESS"/>
                    <SearchDataElement name="SIZE" value='20"'/>
                    <SearchDataElement name="MATERIaAL' value=Steel/>
                </SearchGroupData>
            </IndexItemDetail>
        </IndexItemAdd>
    </IndexItem>
</Index>
```

Supplemental Attribute Examples

The CIF and cXML examples demonstrate how to define supplemental and how to use them in customer catalogs. Both require a type definition file separate from the actual customer catalog file.

This example illustrates the `CatalogItem` type with two supplemental attributes named “Catalog Item Source” and “Hazardous Materials.”

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">
        <OrganizationID>
            <Credential domain="buyersystemid">
                <Identity>1</Identity>
            </Credential>
        </OrganizationID>
    </TypeProvider>
    <Comments> Custom catalog fields for Acme, Inc. </Comments>

      <Type name="CatalogItem" extends="System:CatalogItemInternal">

        <Name xml:lang="en-US">Catalog Item</Name>
        <Name xml:lang="fr-FR">Article de Catalogue</Name>
        <Description xml:lang="en-US">Supplemental fields</Description>

      <TypeAttribute name="Catalog_Item_Source"
     type="Name" isRefinable="yes" isCollection="yes">
            <Name xml:lang="en-US">Catalog Item Source</Name>
            <Name xml:lang="fr-FR">Source d’Article de Catalogue</Name>
        </TypeAttribute>

      <TypeAttribute name="HazardousMaterials"
     type="Id" isRefinable="yes">
            <Name xml:lang="en-US">Hazardous Materials</Name>
            <Name xml:lang="fr-FR">Déchets Dangereux</Name>
        </TypeAttribute>
    </Type>
</TypeDefinition>
```

BMEcat Catalog

To add a custom field, for example, `HazMatClassification` and then add values for the field, you must update the BMEcat file and include the information in the `<USER_DEFINED_EXTENSIONS>` section as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BMECAT SYSTEM "bmecat_new_catalog_1_2.dtd">
<BMECAT version="1.2">
    <HEADER>
```

```
        <GENERATOR_INFO>PSX Toolset v. 1.5</GENERATOR_INFO>
        <CATALOG>
            <LANGUAGE>en</LANGUAGE>
      <CATALOG_ID>C_TESTLIEF</CATALOG_ID>
      <CATALOG_VERSION>001.001</CATALOG_VERSION>
      <CATALOG_NAME>Test catalog BMEcat1</CATALOG_NAME>
      <DATETIME type="generation_date">
                <DATE>2015-08-08</DATE>
                <TIME>09:41:25</TIME>
            </DATETIME>
            <TERRITORY>US</TERRITORY>
            <CURRENCY>USD</CURRENCY>
            <MIME_ROOT>/mime-files/</MIME_ROOT>
            <PRICE_FLAG type="incl_freight">TRUE</PRICE_FLAG>
            <PRICE_FLAG type="incl_packing">TRUE</PRICE_FLAG>
            <PRICE_FLAG type="incl_assurance">FALSE</PRICE_FLAG>
            <PRICE_FLAG type="incl_duty">TRUE</PRICE_FLAG>
        </CATALOG>
        <BUYER>
            <BUYER_ID type="buyer_specific">testbuyer</BUYER_ID>
            <BUYER_NAME>P2PAPC</BUYER_NAME>
        </BUYER>
                <AGREEMENT>
                    <AGREEMENT_ID>Test_1</AGREEMENT_ID>
                    <DATETIME type="agreement_start_date">
                        <DATE>2015-01-01</DATE>
                    </DATETIME>
                    <DATETIME type="agreement_end_date">
                        <DATE>2025-12-31</DATE>
                    </DATETIME>
        </AGREEMENT>
        <SUPPLIER>
      <SUPPLIER_ID type="buyer_specific">sid496</SUPPLIER_ID>
      <SUPPLIER_ID type="supplier_specific">an70000000004</SUPPLIER_ID>
      <SUPPLIER_NAME>JCN Technologies</SUPPLIER_NAME>
        </SUPPLIER>
    </HEADER>
    <T_NEW_CATALOG>
        <CATALOG_GROUP_SYSTEM>
            <GROUP_SYSTEM_ID>Smart Phones</GROUP_SYSTEM_ID>
            <GROUP_SYSTEM_NAME>Smart phones and accessories</GROUP_SYSTEM_NAME>
            <CATALOG_STRUCTURE type="root">
                <GROUP_ID>1</GROUP_ID>
                <GROUP_NAME>Smart phone</GROUP_NAME>
                <PARENT_ID>0</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="node">
                <GROUP_ID>2</GROUP_ID>
                <GROUP_NAME>Sony</GROUP_NAME>
                <PARENT_ID>1</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="node">
                <GROUP_ID>3</GROUP_ID>
                <GROUP_NAME>Samsung</GROUP_NAME>
                <PARENT_ID>1</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="node">
                <GROUP_ID>4</GROUP_ID>
                <GROUP_NAME>iPhone</GROUP_NAME>
                <PARENT_ID>1</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="node">
                <GROUP_ID>5</GROUP_ID>
                <GROUP_NAME>HTC</GROUP_NAME>
                <PARENT_ID>1</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="leaf">
                <GROUP_ID>6</GROUP_ID>
```

```
                <GROUP_NAME>Xperia</GROUP_NAME>
                <PARENT_ID>2</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="leaf">
                <GROUP_ID>7</GROUP_ID>
                <GROUP_NAME>Edge</GROUP_NAME>
                <PARENT_ID>3</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="leaf">
                <GROUP_ID>8</GROUP_ID>
                <GROUP_NAME>Ip6s</GROUP_NAME>
                <PARENT_ID>4</PARENT_ID>
            </CATALOG_STRUCTURE>
            <CATALOG_STRUCTURE type="leaf">
                <GROUP_ID>9</GROUP_ID>
                <GROUP_NAME>OneM9</GROUP_NAME>
                <PARENT_ID>5</PARENT_ID>
            </CATALOG_STRUCTURE>
        </CATALOG_GROUP_SYSTEM>

        <ARTICLE mode="new">
            <SUPPLIER_AID>xperia_z2dd_ngproc</SUPPLIER_AID>
            <ARTICLE_DETAILS>
                <DESCRIPTION_SHORT>xperia z2dd ngproc</DESCRIPTION_SHORT>
                <DESCRIPTION_LONG>Similar to the devices under the Sony Xperia Z
series, the Xperia Z2 has an "Omni-Balance", according to Sony, which is focused
on creating balance and symmetry in all directions. The phone comes with the same
aluminium frame as its predecessor. It also features tempered glass on the front
and back with an aluminium power button placed on the right hand side of the
device</DESCRIPTION_LONG>
                <DELIVERY_TIME>11</DELIVERY_TIME>
                <KEYWORD>Sony</KEYWORD>
                <KEYWORD>Xperia</KEYWORD>
                <KEYWORD>Z2</KEYWORD>
                <KEYWORD>Xperia Z2</KEYWORD>
                <KEYWORD>Ericsson</KEYWORD>
                <ARTICLE_STATUS type="core_article">true</ARTICLE_STATUS>
            </ARTICLE_DETAILS>
            <ARTICLE_FEATURES>
                <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</
REFERENCE_FEATURE_SYSTEM_NAME>
                <REFERENCE_FEATURE_GROUP_ID>44121710</REFERENCE_FEATURE_GROUP_ID>
            </ARTICLE_FEATURES>
            <ARTICLE_ORDER_DETAILS>
                <ORDER_UNIT>C62</ORDER_UNIT>
                <CONTENT_UNIT>C62</CONTENT_UNIT>
                <NO_CU_PER_OU>1</NO_CU_PER_OU>
                <PRICE_QUANTITY>1</PRICE_QUANTITY>
                <QUANTITY_MIN>2</QUANTITY_MIN>
                <QUANTITY_INTERVAL>2</QUANTITY_INTERVAL>
            </ARTICLE_ORDER_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2023-02-02</DATE>
                </DATETIME>
                <DATETIME type="valid_end_date">
                    <DATE>2023-02-18</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>299.99</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>8.19</TAX>
                    <PRICE_FACTOR>0.8</PRICE_FACTOR>
                    <LOWER_BOUND>2</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>149.25</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
```

```
                    <TAX>6.19</TAX>
                    <PRICE_FACTOR>0.8</PRICE_FACTOR>
                    <LOWER_BOUND>5</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>602.15</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>3.19</TAX>
                    <PRICE_FACTOR>0.8</PRICE_FACTOR>
                    <LOWER_BOUND>20</LOWER_BOUND>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
            <ARTICLE_PRICE_DETAILS>
                <DATETIME type="valid_start_date">
                    <DATE>2023-02-02</DATE>
                </DATETIME>
                <DATETIME type="valid_end_date">
                    <DATE>2023-02-18</DATE>
                </DATETIME>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>459.99</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>8.19</TAX>
                    <PRICE_FACTOR>1.0</PRICE_FACTOR>
                    <LOWER_BOUND>2</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>659.25</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>6.19</TAX>
                    <PRICE_FACTOR>1.2</PRICE_FACTOR>
                    <LOWER_BOUND>5</LOWER_BOUND>
                </ARTICLE_PRICE>
                <ARTICLE_PRICE price_type="net_customer">
                    <PRICE_AMOUNT>292.15</PRICE_AMOUNT>
                    <PRICE_CURRENCY>USD</PRICE_CURRENCY>
                    <TAX>3.19</TAX>
                    <PRICE_FACTOR>0.9</PRICE_FACTOR>
                    <LOWER_BOUND>20</LOWER_BOUND>
                </ARTICLE_PRICE>
            </ARTICLE_PRICE_DETAILS>
            <MIME_INFO>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>https://images.pexels.com/photos/248528/pexels-
photo-248528.jpeg</MIME_SOURCE>
                    <MIME_DESCR>Xperia Z2</MIME_DESCR>
                    <MIME_ALT>Front view</MIME_ALT>
                    <MIME_PURPOSE>normal</MIME_PURPOSE>
                    <MIME_ORDER>1</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>https://images.pexels.com/photos/607812/pexels-
photo-607812.jpeg</MIME_SOURCE>
                    <MIME_DESCR>Xperia Z2</MIME_DESCR>
                    <MIME_ALT>Front view</MIME_ALT>
                    <MIME_PURPOSE>thumbnail</MIME_PURPOSE>
                    <MIME_ORDER>1</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>https://images.pexels.com/photos/47261/pexels-
photo-47261.jpeg</MIME_SOURCE>
                    <MIME_DESCR>Xperia Z2</MIME_DESCR>
                    <MIME_ALT>Front view</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>1</MIME_ORDER>
```

```
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>https://images.pexels.com/photos/699122/pexels-
photo-699122.jpeg</MIME_SOURCE>
                    <MIME_DESCR>Black color</MIME_DESCR>
                    <MIME_ALT>Front view black color</MIME_ALT>
                    <MIME_PURPOSE>normal</MIME_PURPOSE>
                    <MIME_ORDER>2</MIME_ORDER>
                </MIME>
                <MIME>
                    <MIME_TYPE>image/jpeg</MIME_TYPE>
                    <MIME_SOURCE>https://images.pexels.com/photos/1092644/pexels-
photo-1092644.jpeg</MIME_SOURCE>
                    <MIME_DESCR>Black color
                    </MIME_DESCR>
                    <MIME_ALT>Front view black color</MIME_ALT>
                    <MIME_PURPOSE>detail</MIME_PURPOSE>
                    <MIME_ORDER>2</MIME_ORDER>
                </MIME>
            </MIME_INFO>
            <USER_DEFINED_EXTENSIONS>
                <EXTENSION>
                    <KEY>HazMatClassification</KEY>
                    <VALUE>GHS01,GHS02,GHS03</VALUE>
                </EXTENSION>
            </USER_DEFINED_EXTENSIONS>
            <ARTICLE_REFERENCE type="accessories">
                <ART_ID_TO>z2_tripod</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="sparepart">
                <ART_ID_TO>z2_headset</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="followup">
                <ART_ID_TO>z2_controller</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="mandatory">
                <ART_ID_TO>z2_case</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="mandatory">
                <ART_ID_TO>z2_wearable2</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="accessories">
                <ART_ID_TO>z2_smartwatch</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="accessories">
                <ART_ID_TO>z2_3dheadset</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="similar">
                <ART_ID_TO>iphone_6s</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="similar">
                <ART_ID_TO>htc_x9</ART_ID_TO>
            </ARTICLE_REFERENCE>
            <ARTICLE_REFERENCE type="similar">
                <ART_ID_TO>samsung_edge</ART_ID_TO>
            </ARTICLE_REFERENCE>
    </ARTICLE>
<ARTICLE mode="new">
    <SUPPLIER_AID>ipowerbankngproc</SUPPLIER_AID>
    <ARTICLE_DETAILS>
        <DESCRIPTION_SHORT>Portable Chargers NGP</DESCRIPTION_SHORT>
        <DESCRIPTION_LONG>A power bank for your iPhone (and other smartphones). The
Vinsic Tulip features a typical design that most power banks have adopted over the
years. It’s small at about 4 inches long. It can fit anywhere quite comfortably.
At 3200mAh, you can charge your iPhone once and some more (~1.5 times). The Vinsic
Tulip comes with a single USB out. The output is 5V at 1A.</DESCRIPTION_LONG>
        <DELIVERY_TIME>22</DELIVERY_TIME>
```

```
        <KEYWORD>ipower</KEYWORD>
    </ARTICLE_DETAILS>
    <ARTICLE_FEATURES>
        <REFERENCE_FEATURE_SYSTEM_NAME>UNSPSC</REFERENCE_FEATURE_SYSTEM_NAME>
        <REFERENCE_FEATURE_GROUP_ID>44121710</REFERENCE_FEATURE_GROUP_ID>
    </ARTICLE_FEATURES>
    <ARTICLE_ORDER_DETAILS>
        <ORDER_UNIT>C62</ORDER_UNIT>
        <CONTENT_UNIT>C62</CONTENT_UNIT>
        <PRICE_QUANTITY>1</PRICE_QUANTITY>
    </ARTICLE_ORDER_DETAILS>
    <ARTICLE_PRICE_DETAILS>
        <ARTICLE_PRICE price_type="net_customer">
            <PRICE_AMOUNT>1301294857.999</PRICE_AMOUNT>
            <PRICE_CURRENCY>USD</PRICE_CURRENCY>
            <TAX>0.3</TAX>
        </ARTICLE_PRICE>
    </ARTICLE_PRICE_DETAILS>
    <MIME_INFO>
        <MIME>
            <MIME_TYPE>image/jpeg</MIME_TYPE>
            <MIME_SOURCE>https://nashome.ariba.com/BMEcat_images/iphone6s/
powerbank.jpeg</MIME_SOURCE>
            <MIME_DESCR>powerbank</MIME_DESCR>
            <MIME_ALT>iphone powerbank</MIME_ALT>
            <MIME_PURPOSE>detail</MIME_PURPOSE>
            <MIME_ORDER>1</MIME_ORDER>
        </MIME>
    </MIME_INFO>
    <USER_DEFINED_EXTENSIONS>
        <EXTENSION>
            <KEY>Hazardous Materials</KEY>
            <VALUE>radioactive,toxic</VALUE>
        </EXTENSION>
    </USER_DEFINED_EXTENSIONS>
</ARTICLE>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>xperia_z2</ART_ID>
    <CATALOG_GROUP_ID>2</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>z2_controller</ART_ID>
    <CATALOG_GROUP_ID>6</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>z2_tripod</ART_ID>
    <CATALOG_GROUP_ID>6</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>z2_headset</ART_ID>
    <CATALOG_GROUP_ID>6</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>z2_case</ART_ID>
    <CATALOG_GROUP_ID>6</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>z2_wearable2</ART_ID>
    <CATALOG_GROUP_ID>6</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>z2_smartwatch</ART_ID>
    <CATALOG_GROUP_ID>6</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>z2_3dheadset</ART_ID>
    <CATALOG_GROUP_ID>6</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
```

```
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>iphone_6s</ART_ID>
    <CATALOG_GROUP_ID>4</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>ipowerbank</ART_ID>
    <CATALOG_GROUP_ID>8</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>iheadset</ART_ID>
    <CATALOG_GROUP_ID>8</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>iusb</ART_ID>
    <CATALOG_GROUP_ID>8</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>icase</ART_ID>
    <CATALOG_GROUP_ID>8</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>ipadapter</ART_ID>
    <CATALOG_GROUP_ID>8</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>imacadapter</ART_ID>
    <CATALOG_GROUP_ID>8</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>samsung_edge</ART_ID>
    <CATALOG_GROUP_ID>3</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>edge_headset</ART_ID>
    <CATALOG_GROUP_ID>7</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>edge_wireless</ART_ID>
    <CATALOG_GROUP_ID>7</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>edge_screenguard</ART_ID>
    <CATALOG_GROUP_ID>7</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>edge_adapter</ART_ID>
    <CATALOG_GROUP_ID>7</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>edge_holder</ART_ID>
    <CATALOG_GROUP_ID>7</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
<ARTICLE_TO_CATALOGGROUP_MAP>
    <ART_ID>htc_x9</ART_ID>
    <CATALOG_GROUP_ID>5</CATALOG_GROUP_ID>
</ARTICLE_TO_CATALOGGROUP_MAP>
    </T_NEW_CATALOG>
</BMECAT>
```

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

ITEMCOUNT: 1
DATA
599238,66056,66056,"Can of hydrogen",11141608,2.95,EA,2,GasesRus,,,
4.95,
      123ABC,true

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 loadmode="Full">
<SupplierID domain="duns">599238</SupplierID>
    <IndexItem>
        <IndexItemAdd>
            <ItemID><SupplierPartID>599238</SupplierPartID></ItemID>
            <ItemDetail>
                <UnitPrice>
                    <Money currency="USD">2.95</Money>
                </UnitPrice>
                <Description xml:lang="en-US">Can of hydrogen</Description>
                <UnitOfMeasure>EA</UnitOfMeasure>
                <Classification domain="UNSPSC">11141608</Classification>
                <ManufacturerPartID>66056</ManufacturerPartID>
                <ManufacturerName>GasesRus</ManufacturerName>
                <Extrinsic name="Catalog_Item_Source">123ABC</Extrinsic>
                <Extrinsic name="HazardousMaterials">true</Extrinsic>
            </ItemDetail>
            <IndexItemDetail>
                <LeadTime>2</LeadTime>
            </IndexItemDetail>
        </IndexItemAdd>
    </IndexItem>
</Index>
```

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.ariba.com

Copyright © 2026 Ariba, Inc. All rights reserved.

This documentation, as well as the Ariba solutions, software and/or services described in it, contain proprietary information. They are provided under a license or other agreement containing restrictions on use and disclosure and are also protected by copyright, patent and/or other intellectual property laws. Except as permitted by such agreement, no part of the document may be reproduced or transmitted in any form by any means, electronic, mechanical or otherwise, without the prior written permission of Ariba, Inc.

Ariba, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the documentation. The information contained in the documentation is subject to change without notice.

Ariba and Ariba products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Ariba, Inc. in the United States and other countries. Please see http://www.ariba.com/legal/trademarks for additional trademark information and notices.

Ariba Sourcing solutions (On Demand and software) are protected by one or more of the following patents, including without limitation: U.S. Patent Nos. 6,199,050; 6,216,114; 6,223,167; 6,230,146; 6,230,147; 6,285,989; 6,408,283; 6,499,018; 6,564,192; 6,871,191; 6,952,682; 7,010,511; 7,072,061; 7,130,815; 7,146,331; 7,152,043;7,225,152; 7,277,878; 7,249,085; 7,283,979; 7,283,980; 7,296,001; 7,346,574; 7,383,206; 7,395,238; 7,401,035; 7,407,035; 7,444,299; 7,483,852; 7,499,876; 7,536,362; 7,558,746; 7,558,752; 7,571,137; 7,599,878; 7,634,439; 7,657,461; 7,693,747; 8,364,577; and 8,392,317. Patents pending.

Other Ariba product solutions are protected by one or more of the following patents:

U.S. Patent Nos. 6,199,050, 6,216,114, 6,223,167, 6,230,146, 6,230,147, 6,285,989, 6,408,283, 6,499,018, 6,564,192, 6,584,451, 6,606,603, 6,714,939, 6,871,191, 6,952,682, 7,010,511, 7,047,318, 7,072,061, 7,084,998; 7,117,165; 7,225,145; 7,324,936; 7,536,362; 8,364,577; and 8,392,317. Patents pending.

Certain Ariba products may include third party software or other intellectual property licensed from a third party. For information regarding software or other intellectual property licensed from a third party, go to http://www.ariba.com/copyrights.cfm.